

ALVL
ASIMUT-(V)LIDORT

Radiative transfer modelling and spectrum retrieval
in a non-scattering atmosphere (ASIMUT) and scattering atmosphere ((V)LIDORT)

User Manual and Science description
(Version 1.2 – Release 11.0 of ASIMUT)

Institut d’Aéronomie Spatiale de Belgique, 3, av. Circulaire
1180 Brussels, Belgium

Abstract.

ASIMUT is a modular program for radiative transfer calculations in planetary atmospheres. The ASIMUT software has been developed to exploit the synergy existing between the growing number of different instruments working under different geometries. The main particularities of the software are:

- (i) The possibility to retrieve columns and/or profiles of atmospheric constituents simultaneously from different spectra, which may have been recorded by different instruments or obtained under different geometries. This allows the possibility to perform combined retrieval, e.g., of a ground based measurement and a satellite-based one probing the same air mass, or from spectra recorded by different instruments on the same platform;
- (ii) The analytical derivation of the Jacobians;
- (iii) The use of the Optimal Estimation method (OEM), using diagonal or full covariance matrices;
- (iv) Its portability;
- (v) Its modularity, hence the ease to add future features.

Initially developed for the Earth atmosphere, its applicability has been extended to extra-terrestrial atmospheres, such as those of Mars and Venus.

ASIMUT has been coupled to SPHER/TMATRIX and (V)LIDORT codes to include the complete treatment of the scattering effects into the radiative transfer calculations.

Table of Contents

1. Introduction.....	6
2. General Description of the software	7
2.1 Implementation	7
2.2 Compiling and running ASIMUT-ALVL.....	10
2.2.1 Pre-processor definitions	10
2.2.2 Compiling under WINDOWS.....	11
2.2.2.1 Create the project.....	11
2.2.3 Compiling under UNIX	14
2.2.4 Linking ASIMUT-ALVL and MATLAB	16
2.2.4.1 Under Windows 32 bits	16
2.2.4.2 Under Windows 64 bits:.....	16
2.2.4.3 Two ways to link ASIMUT and MATLAB	17
2.2.4.4 Under UNIX	23
2.2.5 Using the CLAPACK library	25
2.2.6 Running ASIMUT-ALVL	26
2.2.6.1 Description of the input files	28
2.2.6.2 Description of the ASI file.....	29
2.2.7 Description of the output files.....	30
2.2.7.1 LOG file.....	31
2.2.7.2 General files in /RESULTS	31
2.2.7.3 Specific files in /SAVE.....	34
2.2.8 Examples Provided	35
2.2.8.1 Extracting Line parameters.....	35
2.2.8.2 Nadir IR spectra of Mars atmosphere – no aerosols.....	35
2.2.8.3 Nadir IR spectra of Mars atmosphere – with aerosols.....	35
2.2.8.4 Analysis of a SOIR solar occultation observation.....	36
2.2.9 On-line help	37
2.3 How to.....	38
2.3.1 How to start ASIMUT ?.....	38
2.3.2 How to extract spectroscopic (LP) data from literature database ?	38
2.3.3 How to limit the size of the .LOG file ?	38
2.3.4 How to add a new Instrument ?	38
2.4 Troubleshooting	41
2.4.1 WIN: Debugging information for "project.exe" cannot be found or does not match. Binary was not built with debug information.	41
2.4.2 WIN: No compilation: "Fatal error C1033: cannot open program database "	41
2.4.3 WIN/UNIX: The figures from MATLAB are created but are empty	41
2.4.4 WIN/UNIX: Problem with Files	41
3. Simulation module.....	Error! Bookmark not defined.
3.1 Ray tracing	43
3.1.1 Index of refraction.....	44
3.1.1.1 [Planet] Options	44
3.1.2 Atmospheres	45
3.1.2.1 [AtmosphericModels] options	45
3.1.3 Geometry of observation.....	46
3.1.3.1 [Geometry] options	46

3.2	Optical depths	48
3.2.1	Line-by-line calculation	48
3.2.1.1	Grid and resolution used for the OD simulation.....	51
3.2.1.2	Spectral line catalogues	55
3.2.1.3	Line Mixing	56
3.2.1.4	[Molecules] options.....	57
3.2.1.5	[xxxLP] options	57
3.2.2	Cross sections.....	60
3.2.2.1	Cross sections catalogues	60
3.2.2.2	[xxxXS] options	60
3.2.3	Continua.....	62
3.2.3.1	O ₂ continua	62
3.2.3.2	N ₂ continua	63
3.2.3.3	CO ₂ and H ₂ O continua.....	64
3.2.3.4	[Continua] options.....	64
3.2.4	Rayleigh scattering.....	65
3.2.4.1	[Continua] options.....	65
3.2.5	Aerosols	66
3.2.5.1	[xxxAER] options	68
3.3	Radiative transfer model	70
3.3.1	Determination of the Radiance	70
3.3.2	Atmosphere emission.....	70
3.3.3	Surface contribution.....	71
3.3.4	Determination of the Transmittances.....	72
3.3.5	Brightness temperature	72
3.3.5.1	[Solar] options	72
3.3.5.2	Radiative options in [SPn]	73
3.3.5.3	Radiative options in [SPn_FENn]	73
3.3.6	Instrumental functions	75
3.3.6.1	[Set] options	75
3.3.6.2	[SPn] options	76
3.3.6.3	OPUS instrument	78
3.3.6.4	IMG instrument.....	79
3.3.6.5	ACE instrument.....	79
3.3.6.6	IASI instrument	79
3.3.6.7	SOIR instrument	81
3.3.6.8	[SPn_FENx] options	85
4.	Onion peeling method.....	88
4.1	Onion peeling options	89
5.	Retrieval module.....	90
5.1	General description	90
5.1.1	[Run] options.....	91
5.2	Determination of the Jacobians.....	92
5.3	A priori covariance matrices	92
5.3.1	Covariance options in [xxxLP] , [xxxXS] , and [xxxAER]	93
5.4	Summary of all Variables that can be fitted.....	94
5.4.1	Molecules, cross sections and aerosols densities	94
5.4.2	Variables related to the Spectrum and its windows	94
6.	(V)LIDORT description	96

6.1	Choice between LIDORT and VLIDORT	96
6.2	Aerosols	96
6.2.1	Modified gamma distribution (both codes):	97
6.2.2	Log normal distribution (both codes)	97
6.2.3	Power law distribution (both codes)	97
6.2.4	Gamma distribution (both codes).....	97
6.2.5	Modified power law distribution (both codes).....	97
6.2.6	Bimodal volume log normal distribution (SPHER only).....	97
6.3	Surface	99
6.4	Comment on the Radiance units used in ASIMUT and (V)LIDORT.....	100
6.5	(V)LIDORT input parameters.....	100
6.5.1	Parameters defined in the file.INP	100
7.	Conclusions.....	101
8.	References.....	102

List of Annexes

Annex 1: List of molecules supported and isotopic abundances	106
Annex 2: HITRAN format	110
Annex 3: GEISA format	111
Annex 4: Radiance – irradiance	113
Annex 5: Physical constants	117
Annex 6: Template of file for a new instrument	118
Annex 7: Contacts and rights of use	125

List of Tables

Table 1: Scientific goals fulfilled by the programs included in the ALVL software	8
Table 2: List of pre-processors used by ASIMUT-ALVL.....	10
Table 3: List of calling convention under UNIX (option VLIDORT and VLIDORTF90 are only accessible through the SVN repository of IASB-BIRA)	14
Table 4: Line parameters required by ASIMUT	55
Table 5: Molecules recognized by ASIMUT	60
Table 6: List of BRDF kernels implemented in LIDORT/VLIDORT	99

List of Figures

Figure 1: Scientific programs included in the ALVL software package	7
Figure 2: File structure of the ASIMUT-(V)LIDORT code	9
Figure 3: Several geometries are possible. H_{obs} is the altitude of the observer, ϕ is the zenithal angle, H_{tg} is the tangent height, and H_s is the end point of the ray path.	46
Figure 4: Spectral limits defined and used by ASIMUT	54
Figure 5: Geometry of solar occultation measurements and definition of the onion peeling method.....	88
Figure 6: Example size distributions of Martian dust particles. Size distribution parameters were taken from [Dlugach et al., 2002]......	98

1. Introduction

The number of instruments dedicated to the characterization of the Earth atmosphere composition, either from the ground or from space, even on board balloons or aircraft, is steadily growing. Flexible tools are needed that can be easily adapted to different instruments and different observation geometries. Retrieval is based on accurate forward modelling able to simulate the measurements in the probed air mass and taking into account the characteristics of the instrument (instrumental line function, field of view, ...). Initially developed for nadir looking measurements in the thermal infrared (IR) performed by instruments such as IASI (Infrared Atmospheric Sounding Interferometer) on board METOP-A [1], ASIMUT has been further extended to accommodate different observation geometries and to the simulation of atmospheric spectra for atmospheres other than that of the Earth. For example, ground-based Sun looking measurements recorded by the Fourier transform infrared spectrometer operated by the Belgian Institute for Space Aeronomy (IASB-BIRA) at Uccle or at Ile de la Réunion [2] are exploited to investigate tropospheric and stratospheric composition, but also for satellite validation purposes [3].

IASB-BIRA is also involved in the Venus Express mission of ESA [4], through the SOIR instrument [5, 6]. This instrument, which combines an echelle grating spectrometer and an Acousto-Optical Tunable Filter for the selection of the appropriate grating orders, performs solar occultation measurements in the IR region (2.2-4.3 μm) at a spectral resolution of 0.15 cm^{-1} . As, the techniques involved in the recording and simulation of and in the retrieval from SOIR spectra are similar to those used by instruments dedicated to Earth observation, it was decided to develop the same software to interpret data obtained for different planetary atmospheres. In the case of SOIR data, it consisted in generalizing the ray tracing module to another planet, but also in including the possibility to simulate echelle spectra, in particular the existence of overlapping diffraction orders.

This paper describes the different parts constituting the ASIMUT software. The simulation module consists in a ray tracing unit, units devoted to the calculation of the optical depths and of the radiative transfer, and a final component related to instrumental effects. The retrieval module is based on the Optimal Estimation Method [7] and has the peculiarity to use the Jacobians determined analytically during the forward radiative transfer modelling. The number of runs of the forward model during the retrieval process is therefore much reduced.

Recently the ASIMUT software [8] has been coupled to the (V)LIDORT [9, 10] package, as well as to the SPHER/TMATRIX [11] software to correctly include scattering effects in the Radiative transfer calculations.

In this paper, we present the different blocks defining the new code. The forward modeling as well as retrieval capabilities of ASIMUT will be described and discussed in a series of papers in preparation or already submitted, and related to specific scientific research objectives [12-15].

2. General Description of the software

2.1 Implementation

The ASIMUT program has been written in the C programming language. The modularity of the program has been one of the driving motives to write our own code. Most options driven by the user's choices may be easily extended. It is for example relatively easy to add a new instrument: the only additional information that the user must provide are a function to read the spectra and the instrument line shape function. Another example is the possibility to add other absorption line shapes, besides the ones already implemented.

As one of the objectives was to be able to run the program either on Windows based PCs or on Unix systems, graphical output possibilities are not implemented in the code. However, graphics are possible through the opening of a Matlab session. Data and graphics commands are automatically sent to Matlab while the ASIMUT program is running, if the program has detected the presence of the Matlab environment on the machine.

The user controls the program through a series of text files in which he can specify what has to be calculated, saved and plotted (radiance, transmittance, Jacobians, retrieval results), which spectra are analysed, in which order. It moreover describes the observation geometry corresponding to each spectrum, the spectral intervals, the molecules, cross sections and continua to be taken into account, and all parameters needed for the treatment of the species (isotopes, list of line parameters, file for cross sections, covariance matrices, aerosol characteristics).

The ASIMUT-(V)LIDORT (ALVL) software is a combination of 5 scientific programs and 2 auxiliary blocks written in C and Fortran (see Figure 1 and Table 1). ASIMUT interacts with the (V)LIDORT codes with the help of a special interface used to transfer variables from C to Fortran and vice versa. The aerosol processing SPHER and T-MATRIX codes are linked to the ASIMUT code directly. Input files required for LIDORT/VLIDORT and changes in the ASIMUT input related to the incorporation of the new RT are explained in detail in *ASIMUT_Input_File.pdf*.

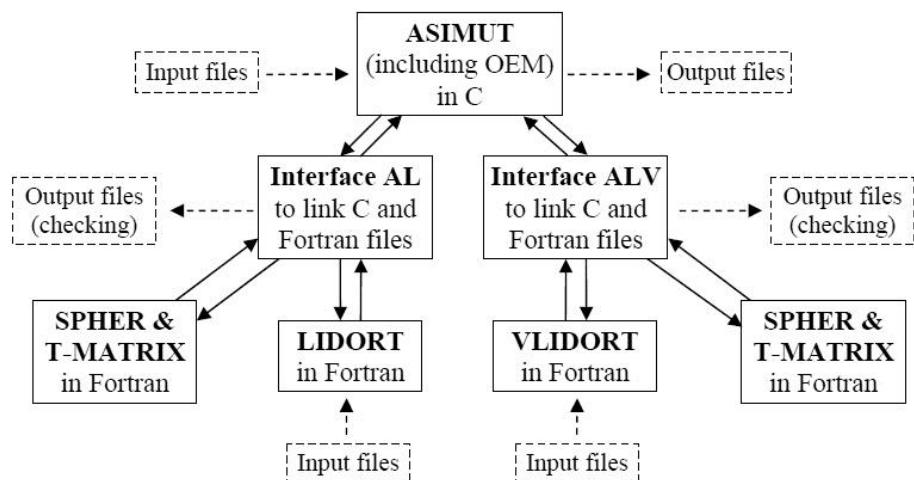


Figure 1: Scientific programs included in the ALVL software package

Program	Purpose
ASIMUT (under Windows and UNIX)	Full independent Radiative Transfer Code (no scattering). When used with (V)LIDORT : Calculation of molecular atmosphere parameters needed as input for (V)LIDORT, simulation of instrument properties, OEM (Optimal Estimation Method).
LIDORT	Full scalar RT, version 3.3
VLIDORT	Full vector RT, version 2.4RT
VLIDORTF90	Full vector RT, version written under F90
SPHER	Calculation of aerosol atmosphere parameters needed as input for (V)LIDORT in case of spherical aerosol particles
T-MATRIX	Calculation of aerosol atmosphere parameters needed as input for (V)LIDORT in case of non-spherical aerosol particles

Table 1: Scientific goals fulfilled by the programs included in the ALVL software

Figure 2 illustrates the file structure of the complete ASIMUT-(V)LIDORT project. The first subdirectory ('ASIMUT') contains all files related to the ASIMUT RT code. It is a self-contained package, and can be compiled and used independently with respect to the complete ASIMUT-(V)LIDORT project. LIDORT and VLIDORT libraries are stored in folders named 'includes', and source files are stored in 'sourcecode' and 'sourcecode_str', respectively. VLIDORT has one additional folder named 'LAPACK_directory' to store the Fortran Lapack-library subroutines. All files and libraries that contribute to aerosol, radiation sources and RT modelling but are not parts of the original LIDORT and VLIDORT packages are designated by the prefix 'aux' and are saved in AUX_SOURCE and AUX_SOURCE_V respectively. VLIDORTF90 is a new addition not shown in this Figure. It will replace the existing VLIDORT older version.

All plots are done through the Matlab interface. All macros ensuring the plotting of input data, of intermediate calculations and of final results are located in the ASIMUT\Matlab subdirectory. They have been compiled to form a Matlab library which is located in

.lib and .so files : ASIMUT\bin\Lib\Win64 or ..\Win32 or ..\UNIX
 all necessary files to include in the ASIMUT\bin\Include\Win64 or ..\Win32 or ..\UNIX respectively
 runtime .dll or .so : ASIMUT\bin\Runtime\Win64 or ..\Win32 or ..\UNIX

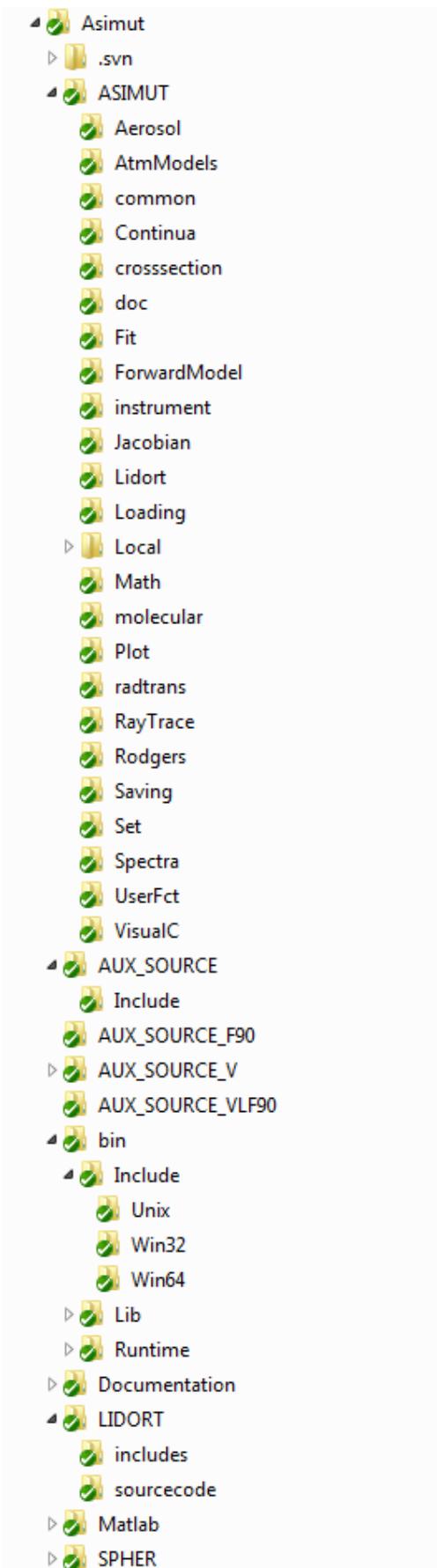


Figure 2: File structure of the ASIMUT-(V)LIDORT code

2.2 Compiling and running ASIMUT-ALVL

2.2.1 Pre-processor definitions

The following pre-processor definitions are used by ASIMUT-ALVL:

WIN32	Indicates that the program is compiled and run on a Windows 32 machine.
WIN64	Indicates that the program is compiled and run on a Windows 64 machine (default)
UNIX	to indicate lines only used under UNIX (calls to LIDORT, VLIDORT, SPHER/TMATRIX)
ASIMUT	to specify that only ASIMUT has to be considered
LIDORT	to specify if ASIMUT has to be linked with LIDORT
VLIDORT	to specify if ASIMUT has to be linked with VLIDORT
VLIDORTF90	to specify if ASIMUT has to be linked with VLIDORT F90
LIDORTF90	to specify if ASIMUT has to be linked with LIDORT F90
DEBUG	when running under DEBUG configuration. In that case all in/out from functions are written in the DebugFile
MATLAB [*]	to indicate that all calls to MATLAB functions will be done through the use of the MATLAB Engine. This option is to be selected when MATLAB is installed on the computer but without the runtime library.
MATLABRUNTIME [*]	To indicate that the calls to MATLAB will be done through the runtime library of MATLAB and a specific ASIMUT library containing all plotting functions
OPERATIONAL	Operational mode: no messages, no files are saved, no plots are possible. Only the file .OUT is created with almost full content. The LOG file contains only final results.
FULLLINEMIXING	Line mixing calculation using the 1 st approximation does not require any specific action from the USER at compile/linking time. However, if the user wants to use the ‘full’ line mixing approach, the LAPACK library need to be included into the project. The Default option is ‘1 st ORDER LINE MIXING’ which does not require the inclusion of any additional library. Please refer to Section 2.2.5 for more information about how to include the LAPACK library. See also the <i>[xxxxLP]LineMixing</i> option.

Table 2: List of pre-processors used by ASIMUT-ALVL.

^{*} If neither MATLAB nor MATLABRUNTIME are present, all calls to MATLAB plotting functions are non-functional. This is the default option (no MATLAB). See Section on ‘how to link ASIMUT and MATLAB’ (Section 2.2.4) for more details.

2.2.2 Compiling under WINDOWS

Under Windows, only the functionalities of ASIMUT are available, no links to those of (V)LIDORT or SPHER/TMATRIX are possible. The physical processes thus encompass only non-scattering atmospheres. Aerosols are included but only as extinctions. The lines of code corresponding to calls to (V)LIDORT or SPHER/TMATRIX functions are between pragma :

```
#ifdef (_UNIX_)
#endif
```

or

```
#ifdef (_LIDORT_) [or #ifdef (_VLIDORT_)]
#endif
```

The programme compiles without any problem, but none of the calls to (V)LIDORT(F90) or SPHER/T-MATRIX functions are possible. In that case, only the option ‘Code=ASIMUT’ in the input ASI file is working properly (see *ASIMUT_Input_File.pdf* for the complete description of all control parameters accepted by ASIMUT-(V)LIDORT).

The package contains the project definition defined under MS Visual C++ 2010 (*ASIMUT\ASIMUT\VisualC\Asimut.sln*). However, the code does not contain any call to functions specific to the Visual C++ compiler and it should be possible to compile it with any other compiler.

Two dedicated files explain in details how to install ASIMUT on a PC running under Windows: *ASIMUT_Screenshots_Install.pdf* and *ASIMUT_FileExtension.pdf*.

Special care has to be taken to ensure the link between ASIMUT and Matlab to be effective (see §2.2.4).

2.2.2.1 Create the project

Create a new sub-directory

You can either use an existing sub-directory (such as the ‘VisualC’) or create a new one [ex: ...\\ASIMUT\\ASIMUT\\Local]

Open the Visual C++ environment (VC2008 or VC2010)

Be sure to be Administrator of your PC or to start VisualC with the option ‘Run as Administrator’ (for VC2010, no such problem encountered with VC2008).

Create a new project based on existing files

Select ‘File’ → ‘New’ → ‘Project from existing files’

You are then proposed a series of screens in which step by step you will define the new project

Screen 1 – What type of project → Visual C++

Screen 2 – Project File location → Browse to and select the ...\\ASIMUT\\ASIMUT\\Local

Project Name → Asimut

Click on the Add Button and select the ...\\ASIMUT\\ASIMUT directory; deselect the ...\\ASIMUT\\ASIMUT\\Local directory; check that the option 'Add files to the project from these folders' is selected

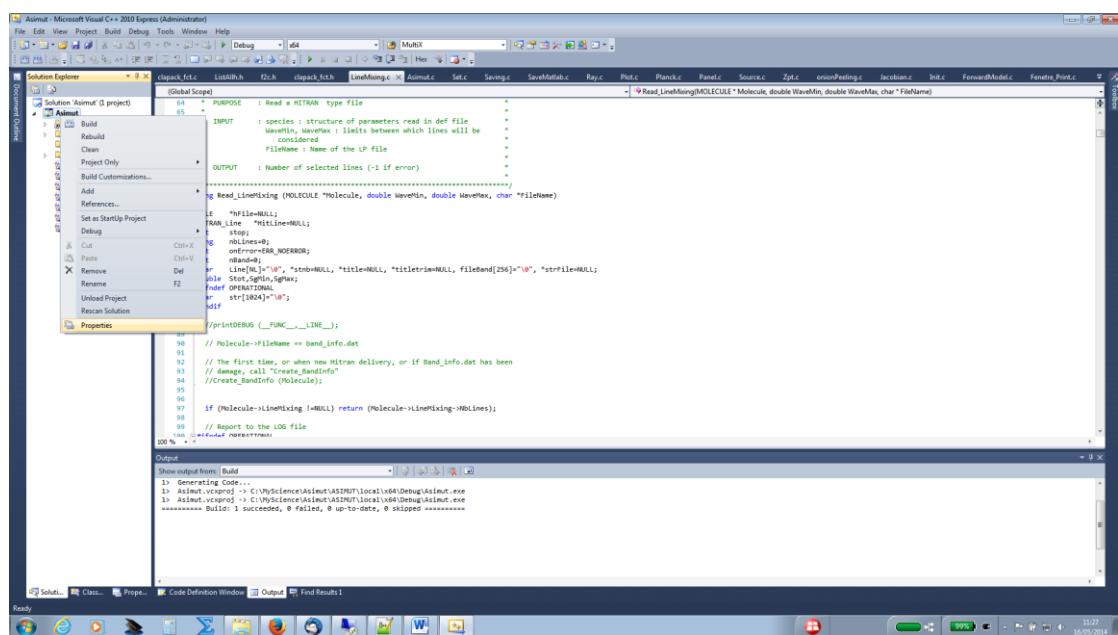
In the field 'File types to add to the project': leave only '*.c;*.h'

Screen 3 – Project Type → 'Console Application project'

Screen 4 – Finish

Modify the properties of the new project

Right-click on 'Asimut' in the frame 'Solution Explorer' [or go to the menu 'Project' → 'Properties']



First, at the top of the window, select the option 'All configurations' in the 'Configuration' list

In the 'Configuration Properties' section:

Debugging

Command arguments → 'path\\file.asi' [If you intend to run the program within the Visual C environment]

C/C++

Preprocessor

Preprocessor Definitions → Add the following:

WIN32 (or WIN64);

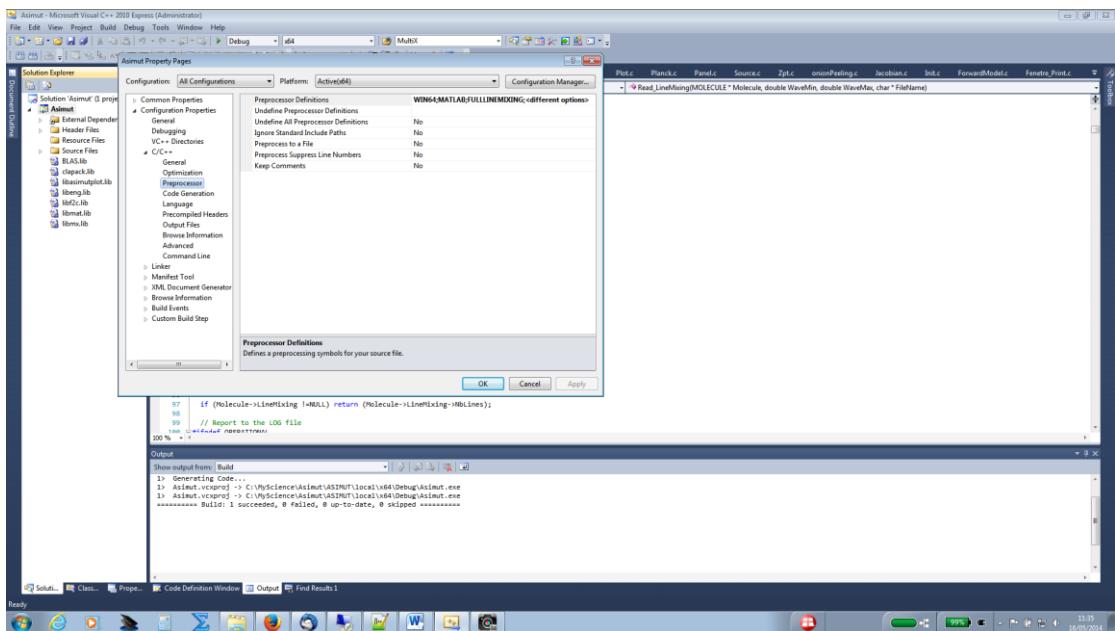
_CRT_SECURE_NO_WARNINGS;

DEBUG (if you want to use the Debug options of Asimut, those are not the same as the Debug options provided by Visual C)

+ other options as needed (MATLAB,

MATLABRUNTIME, FULLLINEMING,
OPERATIONAL)

+ deselect the option 'Inherit from parent'



If you intend to use Matlab or the LAPACK library

If you need one of these options, you must specify where to find the *.h, *.lib, (for the compilation) and *.dll or *.so (for the run).

Please refer to Section 2.2.4 and 2.2.5 for additional inputs and references. By default all links to MATLAB are deselected and the LAPACK library is not needed (only 1st order line mixing).

2.2.3 Compiling under UNIX

A *makefile* is provided: it produces 4 different executable files: ASILID, asimut, AVL or AVL_F90, depending on the command line options (see Table 3). !

The version of the software package for outside IASB-BIRA only includes the possibility to link ASIMUT with LIDORT : only the ‘USE=ASIMUT’ and ‘USE=LIDORT’ options are available. For outside IASB-BIRA the makefile provided is called ‘*Makefile_ToGive*’.

USE=LIDORT	will compile and link the LIDORT code → <i>ASILID</i> executable (Default)
USE=ASIMUT	will compile ONLY the ASIMUT part of the code → ‘ <i>asimut</i> ’ executable
USE=VLIDORT	<i>will compile and link the VLIDORT code → AVL executable [not available outside IASB-BIRA]</i>
USE=VLIDORTF90	<i>will compile and link the VLIDORT F90 code → AVL_F90 executable [not available outside IASB-BIRA]</i>
MATLAB = NO / YES	Allows to indicate if compilation will be done with or without MATLAB embedded. (NO is the Default value)
MATLABRUNTIME= YES / NO	Allows to indicate if the compilation/linking will be done with the MATLAB Runtime library. If NO is selected, then the links to MATLAB are done through the MATLAB Engine. (YES is the Default value)
OPERATIONAL= YES / NO	Allows to indicate the value of the pre-processor OPERATIONAL. (NO is the Default value)
FULLLINEMIXING= YES / NO	Allows to indicate the value of the pre-processor FULLLINEMIXING. (NO is the Default value) !! Full treatment of the Line Mixing is not yet implemented under UNIX (only 1 st approximation)
CC=icc /gcc	Specify the C compiler (Default=icc)
FC = ifort /gfortran	Specify the Fortran compiler (Default=ifort)
clean_c	Remove all *.o file coming from *.c files
clean_f	Remove all *.o file coming from *.f or *.f90 files (in LIDORT, VLIDORT, VLIDORTF90, AUX_SOURCE, AUX_SOURCE_V and AUX_SOURCE_VLF90)
clean_bak	Remove all *.bak files
clean	Remove all *.bak, *.o files

Table 3: List of calling convention under UNIX (option VLIDORT and VLIDORTF90 are only accessible through the SVN repository of IASB-BIRA)

Examples of usage :

Within IASB-BIRA:

to compile using the LIDORT option with MATLAB, and the `icc` and `ifort` compilers :

```
> make USE=LIDORT MATLAB=YES
```

to compile using only ASIMUT, no MATLAB, using the `gcc` and `gfortran` compilers:

```
> make USE=ASIMUT CC=gcc FC=gfortran
```

Outside IASB-BIRA:

to compile using the LIDORT option:

```
> make -f Makefile_ToGive USE=LIDORT
```

to compile using only ASIMUT:

```
> make -f Makefile_ToGive USE=ASIMUT
```

Do not forget to modify the makefile to correspond to your own system. You can specify the C and Fortran compilers that will be used. For the moment the makefile supports `gcc`, `icc`, `ifort` and `gfortran`.

Change also the Path indicating the place where the Matlab libraries can be found. See also Section 2.2.4 for more details on the linking to MATLAB.

Even if the code has been compiled with the `_LIDORT_`, `_VLIDORT_`, or `_VLIDORTF90_` options, it is possible to run the software with all (V)LIDORT functionalities disabled (pure ASIMUT code) by forcing ‘`Code=ASIMUT`’ in the input `.ASI` file. Please beware, the contrary is not true : compiling with the `_ASIMUT_` option, will disable all (V)LIDORT calls. Likewise compiling with the `_LIDORT_` option, will deactivate the VLIDORT and VLIDORTF90 calls (and similarly with `_VLIDORT_` and `_VLIDORTF90_`, the calls to the other options, except only ASIMUT will be disabled).

All tests have been performed on the LIDORT version. IT IS HIGHLY RECOMMENDED TO USE ONLY THE `USE=LIDORT` OPTION (or `ASIMUT` if no scattering is needed).

Only the LIDORT code is provided outside IASB-BIRA, VLIDORT and VLIDORTF90 are only accessible through the IASB-BIRA svn repository.

2.2.4 Linking ASIMUT-ALVL and MATLAB

All plots are done through a MATLAB interface. Therefore, the link between ASIMUT-ALVL and MATLAB should be ensured. The link has been successfully tested under Windows (32 and 64) and UNIX, but some limitations have been found, in relation to the compatibility existing between Visual C++ and MATLAB versions.

2.2.4.1 Under Windows 32 bits

No problems encountered, neither with VC 2008, nor with VC 2010. Be sure that the WIN32 precompiler is set, and the NOMATLAB one is not present (except if you do not want to use Matlab or if Matlab is not installed on your PC).

2.2.4.2 Under Windows 64 bits:

There seems to be an incompatibility between the new DLLs furnished by MATLAB and **VC 2008**. Please compile/link with the Preprocessor Definition NOMATLAB. The program then works fine, but, of course, no links to MATLAB are possible.

When using **VC 2010**, follow the next steps:

1. Download and install the Windows Software Development Kit version 7.1.
Visual C++ 2010 Express does not include a 64 bit compiler, but the SDK does.
A link to the SDK:
<http://msdn.microsoft.com/en-us/windowsserver/bb980924.aspx>
2. Choose the 'X64' project configuration. This is obtained by :
 - a. *Changing the project configuration.*
Go to Properties of your project. On the top of the dialog box there will be a "Configuration" drop-down menu. Make sure that selects "All Configurations." There will also be a "Platform" drop-down that will read "Win32." Finally on the right there is a "Configuration Manager" button - press it. In the dialog that comes up, find your project, hit the Platform drop-down, select New, then select x64. Now change the "Active solution platform" drop-down menu to "x64." When you return to the Properties dialog box, the "Platform" drop-down should now read "x64";
 - b. *Changing the toolset.*
In the Properties menu of your project, under Configuration Properties | General, change Platform Toolset from "v100" to "Windows7.1SDK";

3. Be sure to add WIN64 preprocessor definition, and to remove the NOMATLAB one.
4. Be sure either to use a user ADMIN account or to start VC 2010 with the option “Run as Administrator”. Then first launch VC 2010 and then import the *Asimut_vc2010_Win64.sln* project. Do not double click on the project file.

2.2.4.3 Two ways to link ASIMUT and MATLAB

There are two ways to link ASIMUT and MATLAB: the first one uses the Matlab Engine and the second one uses the runtime Matlab library.

The choice between the 2 ways is based on the version of Matlab that has been installed on your PC. In all cases, the libraries needed to use the Matlab Engine are present; in some cases the runtime library is also installed. Once you have checked which installation is present on your system, you need to tell ASIMUT which version to use:

TIP :

If the Matlab Runtime library is installed on your system, then you should find a sub-directory like

C:\Program Files\MATLAB\R2007b\runtime ==[MATLAB_runtime]

Containing the *mclmcrrt8_1.dll* [the numbers might change depending on your version of Matlab].

Use the MATLAB precompiler if only the Engine is installed → Method 1 [Engine]

Use the MATLABRUNTIME precompiler if the Matlab Runtime library is installed → Method 2 [Runtime]

Method 1 [Engine]

Step 1: Add the MATLAB Engine libraries

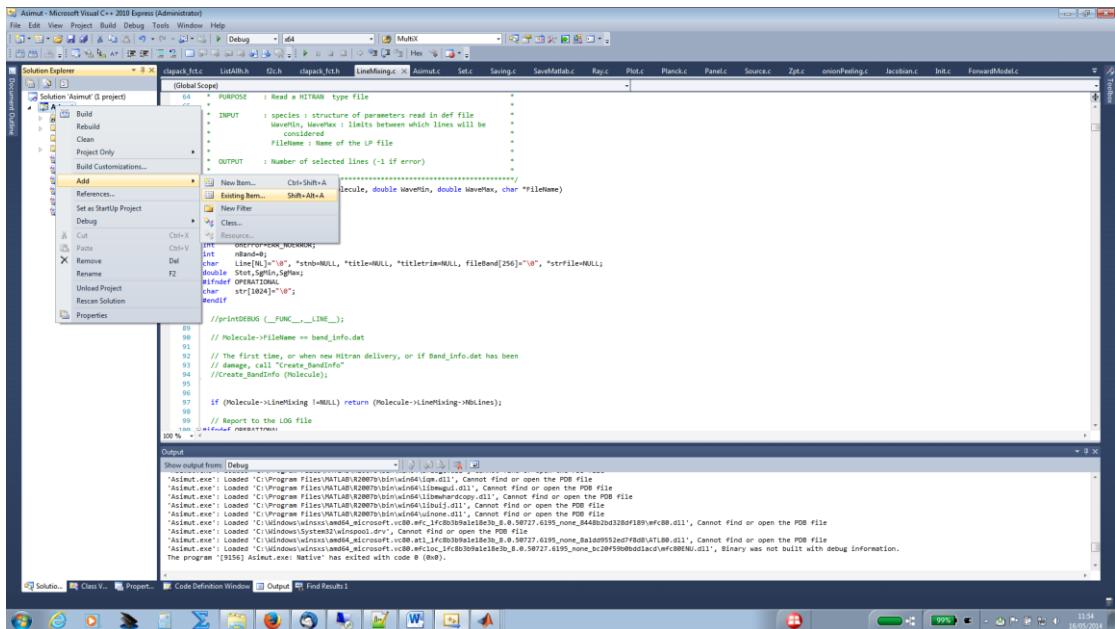
Be sure to “Add” the 3 following libraries “*libeng.lib*”, “*libmat.lib*”, and ‘*libmx.lib*’ from the *correct MATLAB* sub-directory containing the libraries for external use. This should be like :

C:\Program Files\MATLAB\R2007b\extern\lib ==[MATLAB_extern_lib]

1/ Import the libraries into the project

In the “Solution Explorer”,
 right-click on “Asimut”
 choose “Add > Existing Item...”,
 go to the *[MATLAB_extern_lib]* directory and

select the “*libeng.lib*”, “*libmat.lib*”, and ‘*libmx.lib*’ libraries.



2/ Specify to the compiler where to find the libraries and the related ‘Include’

In the “Solution Explorer”,

right-click on “Asimut”

select “Properties”

Under “Configuration Properties > VC++ Directories”

Select “Include Directories” > “<Edit>”

Add the line:

C:\Program Files\MATLAB\R2007b\extern\include == [MATLAB_Include]

In the “Solution Explorer”,

right-click on “Asimut”

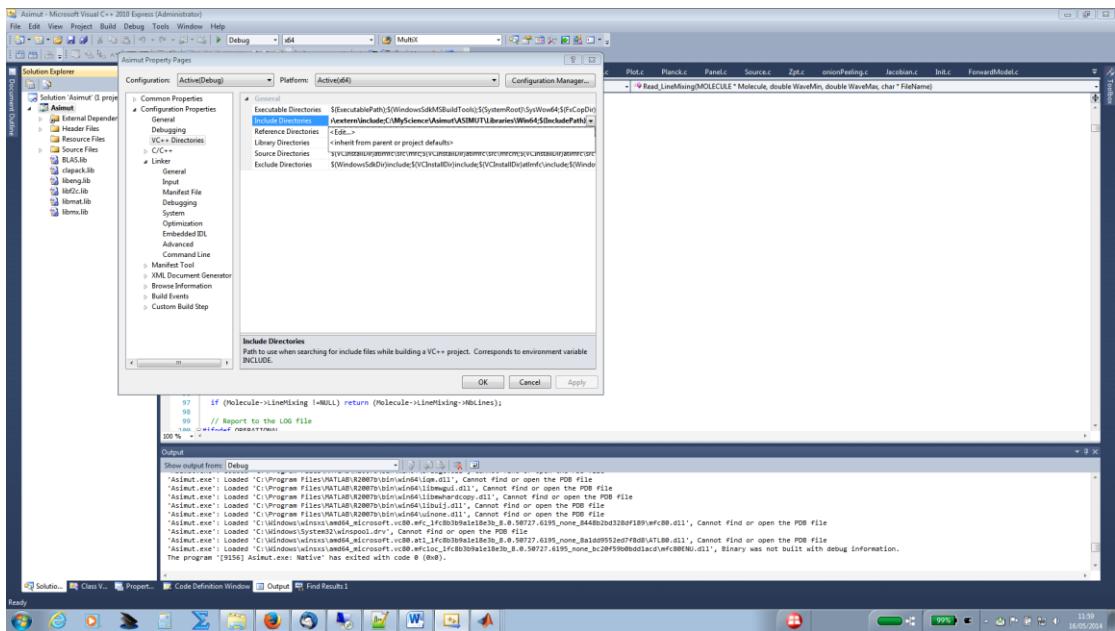
select “Properties”

Under “Configuration Properties > VC++ Directories”

Select “Library Directories” > “<Edit>”

Add the line:

C:\Program Files\MATLAB\R2007b\extern\lib == [MATLAB_extern_lib]

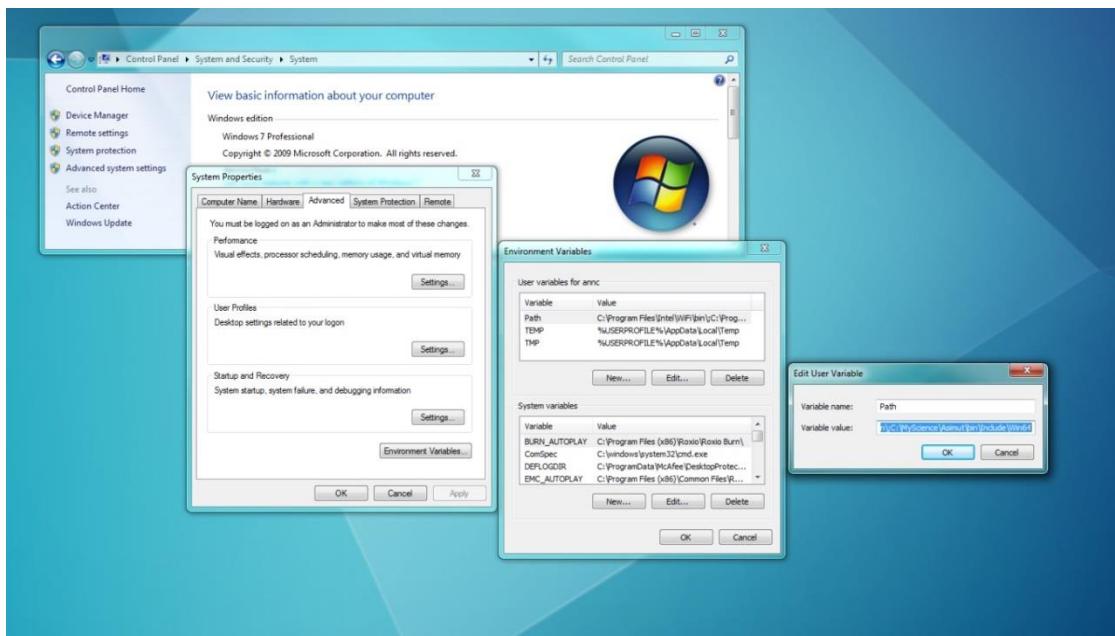


3/ Specify to the compiled ASIMUT.EXE where to find the dll related to the libraries

The easiest method is to change the Environment Settings, including into the definition of PATH, where to find the different dll. You need to add the following sub-directory:

C:\Program Files\MATLAB\R2007b\bin\win64 ==[MATLAB_bin]

\$(Path)=\$(Path); [MATLAB_bin]



Another method consists in changing the environment within VISUAL C++.

In the “Solution Explorer”,

right-click on “Asimut”

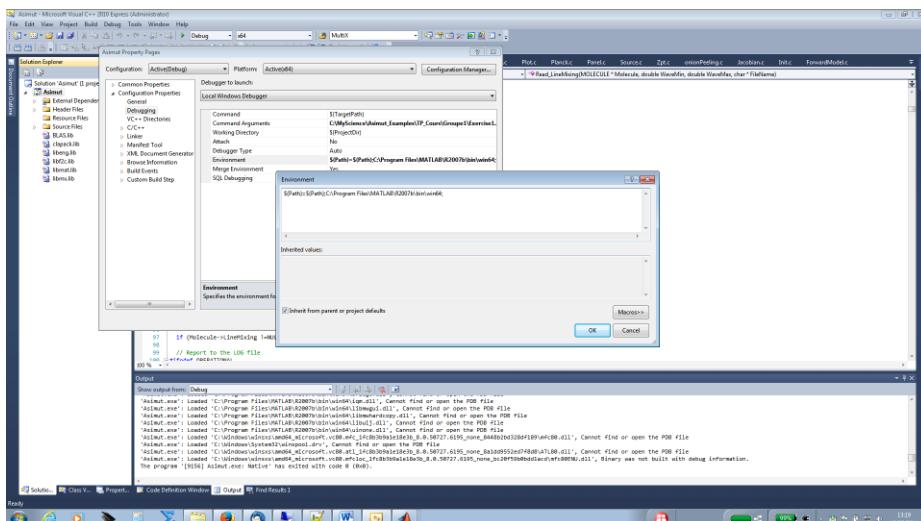
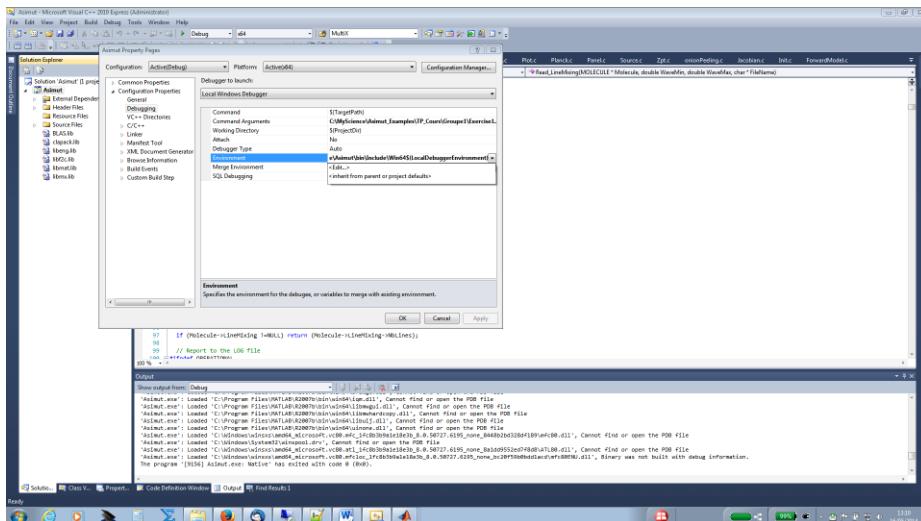
select “Properties”

Under “Debugging”

Select “Environment” > “<Edit>”

Add the directory to the PATH :

$\$(Path)=\$(Path);C:\Program Files\MATLAB\R2007b\bin\win64$



Step 2: Indicate where the MATLAB macros are located

The second step to ensure the link is adding the path to the MATLAB macros (ASIMUT\Matlab) to the PATH variable of MATLAB. This link can also be done through the [Directories]dirMatlab in the ASI control file (see ASIMUT_Input_File.pdf for more details).

Method 2 [Runtime]

Be sure to “Add” the 2 following libraries “*mclmcr.lib*”, and the “*libasimutplot.lib*”. The first one (“*mclmcr.lib*”) is the Matlab Runtime library to be found in the *[MATLAB_extern_lib]* sub-directory :

C:\Program Files\MATLAB\R2007b\extern\lib ==*[MATLAB_extern_lib]*

The second one is the ASIMUT library containing all plotting functions. It is located in the *[ASIMUT_Lib]* sub-directory:

Asimut\bin\Lib\Win64 ==*[ASIMUT_Lib]*

and its related *.h in

Asimut\bin\Include\Win64 ==*[ASIMUT_Include]*

1/ Import the libraries into the project

In the “Solution Explorer”,

right-click on “Asimut”

choose “Add > Existing Item...”

(1) go to the *[MATLAB_extern_lib]* directory and select the “*mclmcr.lib*” library.

(2) repeat and go to the *[ASIMUT_Lib]* directory and select the “*libasimutplot.lib*” library.

2/ Specify to the compiler where to find the libraries and the related ‘Include’

In the “Solution Explorer”,

right-click on “Asimut”

select “Properties”

Under “Configuration Properties > VC++ Directories”

Select “Include Directories” > “<Edit>”

Add the 2 lines:

C:\Program Files\MATLAB\R2007b\extern\include == *[MATLAB_Include]*

Asimut\bin\Include\Win64 ==*[ASIMUT_Include]*

In the “Solution Explorer”,

right-click on “Asimut”

select “Properties”

Under “Configuration Properties > VC++ Directories”

Select “Library Directories” > “<Edit>”

Add the 2 lines:

C:\Program Files\MATLAB\R2007b\extern\lib ==*[MATLAB_extern_lib]*

Asimut\bin\Lib\Win64 ==*[ASIMUT_Lib]*

3/ Specify to the compiled ASIMUT.EXE where to find the dll related to the libraries

The easiest method is to change the Environment Settings, including into the definition of PATH, where to find the different dll. You need to add the 2 following sub-directories:

*C:\Program Files\MATLAB\R2007b\runtime ==[MATLAB_runtime]
Asimut\bin\Runtime\Win64 ==[ASIMUT_dll]*

\$(Path)=\$(Path); [MATLAB_runtime]; [ASIMUT_dll]

Another method consists in changing the environment within VISUAL C++.

In the “Solution Explorer”,

right-click on “Asimut”

select “Properties”

Under “Debugging”

Select “Environment” > “<Edit>”

Add the 2 directories

\$(Path)=\$(Path); [MATLAB_runtime]; [ASIMUT_dll]

2.2.4.4 Under UNIX

Change the following line in the makefile to indicate where the libraries can be found

```
MATLABLIBPATH=/opt/matlab/bin/glnxa64
MATLABRUNPATH=/opt/matlab/runtime/glnxa64
MATLABINCPATH=/opt/matlab/extern/include
```

and

```
ASIMUTLIBPATH=bin/Lib/Unix
ASIMUTINCPATH=bin/Include/Unix
```

As under WINDOWS, the user can choose between two ways of importing the MATLAB functionalities: through the Engine, or through the Matlab Runtime library.

When using the Engine: Use MATLABRUNTIME=NO on the command line for the compilation

To ensure direct communication between ASIMUT and Matlab under UNIX, the user can use the *[Directories]dirMatlab* in the ASI control file (see *ASIMUT_Input_File.pdf* for more details).

When using the Runtime Library method: Use MATLABRUNTIME=YES on the command line for the compilation

Makefile (first lines)

```
#####
# Makefile for compilation of the ASIMUT-LIDORT/VLIDORT package #
#      created by V.Letocart, BIRA, Belgium on 07/08/2009      #
#      last updated : 14/05/2014      #
#####

# calling examples:
# to compile using the LIDORT option:
# with Matlab:
# > make USE=LIDORT MATLAB=YES
# without Matlab:
# > make USE=LIDORT MATLAB=NO
#
# to compile using only ASIMUT:
# with Matlab:
# > make USE=ASIMUT MATLAB=YES
# without Matlab, but using gcc, and gfortran :
# > make USE=ASIMUT MATLAB=NO CC=gcc FC=gfortran

# !!! do not forget to use 'make clean' when you change the compiling option

# indicate here the C and Fortran compilers
# -----
# CC supported : icc, gcc
# FC supported : ifort, gfortran
#
# if you use other compilers, do not forget to define the options
#   CFLAGS for CC
#   FFLAGS for FC
#

# Default values
# -----
CC = icc
```

```

FC = ifort
USE = LIDORT
MATLAB = NO
MATLABRUNTIME = YES

# indicate here where the MATLAB libraries are located
# -----
MATLABLIBPATH=/opt/matlab/bin/glnxa64
MATLABRUNPATH=/opt/matlab/runtime/glnxa64
MATLABINCPATH=/opt/matlab/extern/include
ASIMUTLIBPATH=bin/Lib/Unix
ASIMUTINCPATH=bin/Include/Unix

ifeq ($(MATLAB), NO)
  MCFLAGS = -DNOMATLAB
endif
ifeq ($(MATLAB), YES)
  ifeq ($(MATLABRUNTIME), NO)
    MCFLAGS = -DMATLAB -I$(MATLABINCPATH) -I$(ASIMUTINCPATH)
    MFFLAGS = -L$(MATLABLIBPATH) -lmx -lmat -leng -lut -Wl,-
rpath=$(MATLABLIBPATH)
  else
    MCFLAGS = -DMATLABRUNTIME -I$(MATLABINCPATH) -I$(ASIMUTINCPATH)
    MFFLAGS = -L$(MATLABLIBPATH) -L$(MATLABRUNPATH) -L$(ASIMUTLIBPATH) -
lasimutplot -lmwmclmcrrt -Wl,-rpath=$(MATLABLIBPATH)
  endif
endif

COBJECTS = $(patsubst %.c,%.o,$(wildcard ASIMUT/*/*.c ))
...

```

2.2.5 Using the CLAPACK library

Line mixing has been included for CO₂ (and CH₄, in near future). Two methods of calculations have been implemented, the first one is called ‘1st order approximation’ and the second one is the full mixing method (see REF for more details on the two methods). When the full mixing method is selected, the CLAPACK needs to be included into the ASIMUT-ALVL project.

For the moment, this is only possible with the version of ASIMUT-ALVL compiled under WINDOWS.

The different files are located in:

<i>Asimut\bin\Lib\Win64 (or \Win32)</i>	<i>==[ASIMUT_Lib]</i>
<i>Asimut\bin\Include\Win64 (or \Win32)</i>	<i>==[ASIMUT_Include]</i>

To include the CLAPACK libraries:

1/ Import the libraries into the project

In the “Solution Explorer”,
right-click on “Asimut”
choose “Add > Existing Item...”,
go to the *[ASIMUT_Lib]* directory and select the “*BLAS.lib*”, “*clapack.lib*”, and “*libf2c.lib*” files.

2/ Specify to the compiler where to find the libraries and the related ‘Include’

In the “Solution Explorer”,
right-click on “Asimut”
select “Properties”
Under “Configuration Properties > VC++ Directories”
Select “Include Directories” > “<Edit>”
Add the line:
Asimut\bin\Include\Win64 ==[ASIMUT_Include]

In the “Solution Explorer”,
right-click on “Asimut”
select “Properties”
Under “Configuration Properties > VC++ Directories”
Select “Library Directories” > “<Edit>”
Add the line:
Asimut\bin\Lib\Win64 ==[ASIMUT_Lib]

2/ Specify to the compiler the FULLLINEMIXING option

Be sure to add the FULLLINEMIXING pre-processor option.

2.2.6 Running ASIMUT-ALVL

Under Windows, if you are not using the Visual C environment, WINDOWS may require some additional software to be installed to be able to run executable programs. You will then need to download and install the Microsoft .NET Framework package. The latest version can be downloaded from the microsoft website (<http://www.microsoft.com/net/>).

The calling convention is:

```
> Asimut X:/fullpath/FileASI.asimut
```

where *Asimut* is the executable. The call being done in the directory where asimut.exe is located, if not you need to specify the path where this file has been saved.

Do not forget to specify where to find the dll used by the project if you want to use the functionalities of MATLAB (see 2.2.4 for more details).

Under UNIX, and if using the makefile provided with the project, this executable is named ‘asimut’, ‘ASILID’ or ‘AVL’ when using the ASIMUT, LIDORT and VLIDORT options respectively.

```
> ./asimut X:/fullpath/FileASI.asimut
> ./ASILID X:/fullpath/FileASI.asimut
> ./AVL X:/fullpath/FileASI.asimut
```

A script is also provided “*run_alvl.sh*” which is particularly useful when using the plotting functionalities, which require the specification of the directories where the *.so can be found.

The call to the script is the following:

```
> ./Run_alvl.sh EXENAME ASI_file
```

Where *EXENAME* is the name of the executable to use (asimut, ASILID, AVL, ...) and *ASI_file* is the first input file needed by ASIMUT-ALVL. The script automatically points to the right library path.

```
Run_alvl.sh
#!/bin/bash

export MATLABLIBPATH=/opt/matlab/bin/glnxa64
export MATLABRUNPATH=/opt/matlab/runtime/glnxa64
export JAVALIBPATH=/opt/matlab/sys/java/jre/glnxa64/jre/lib/amd64/server
export ASIMUTRUNPATH=`dirname $0`/bin/Runtime/Unix

if [ "$1" = "-h" ] || [ "$1" = "--help" ]; then
    echo "$0: run ALVL binary" >&2
    echo "Usage: $0 NAME_OF_EXE ASI_file" >&2
    echo "  NAME_OF_EXE: ASILID or ASILIDF90 or asimut" >&2
    exit 0
fi
```

```
export
LD_LIBRARY_PATH="$MATLABLIBPATH":'$MATLABRUNPATH":'$ASIMUTRUNPATH":'$JAVALIB
PATH":$LD_LIBRARY_PATH

exec `dirname $0`/"$1" "$2"
```

Several output files are created depending on what the user has asked. One file is **ALWAYS** created is a LOG file which contains information on the run of ASIMUT. It contains also the major results and if the program stopped, why and where in the code it failed. Reading this file in case of problem, gives in general the reason of the problem encountered. See also Section “Troubleshooting” for more information on Error Termination with ASIMUT.

2.2.6.1 Description of the input files

ASIMUT-ALVL uses a series of input files: **file.ASI** and **file.INP**, when calling only ASIMUT. When using (V)LIDORT, an additional file is mandatory (**file_LIDORT_flags**). If calls are made to SPHER/TMATRIX then a further file describing the aerosols has to be provided (**file_aerosol.INP**).

The extensions of these files are not imposed and can be changed. However in the following we will consider ASI and INP the definition of the two types of files required by ASIMUT.

The *ASIMUT_Input_File.pdf* file describes in details the formats and contents of the two input files used or needed by ASIMUT-ALVL. The first file (*.asi or *.asimut, by convention) contains a series of parameters definitions for the general run conditions. For ASIMUT, a second file (*.inp by convention) contains the details of the simulations or retrievals to be performed. When calling ASIMUT-ALVL, a specific file containing LIDORT parameters is also needed. And when calling SPHER/TMATRIX a file is needed to describe the aerosol included.

In this section we will briefly describe the ASI file. The content of the INP file will be described in the next sections in parallel to the description of the physics and mathematics of the algorithms.

All files are constructed on the same canvas. Values of parameters are specified by lines such as:

Parameter = *value*

The ‘parameter’ name must be a recognized name and the possibilities are described in the following and listed in *ASIMUT_Input_File.pdf*. Spaces (and tabs) are automatically removed, so that

Parameter = *value*
is equivalent to
Parameter=*value*

Parameters’ names are case insensitive :

PARAmeter = *value*
is equivalent to
parameter = *value*

Series of related parameters are regrouped under different Section, whose names are indicated within brackets: [Section]. For example, all parameters dealing with the characterization of the solar spectrum are located under the [Solar] section.

Within one section, the order of appearance of the Parameters is not important.

[*Section*]
PARAmeterA = *value*
PARAmeterB = *value*
PARAmeterC = *value*

is equivalent to

```
[Section]
PARAmeterC = value
PARAmeterA = value
PARAmeterB = value
```

The ‘%’ symbol at the start of the line is used for comments and to de-select some parameter definitions. For example,

```
%parameter = value
```

will not be considered.

Almost all parameters have DEFAULT values, taken if the corresponding line is not present in the ASI or INP files. See *ASIMUT_Input_File.pdf* for the full definition of the possible parameters and their possible and default values.

2.2.6.2 Description of the ASI file

[Run]Verbose : defines the level of comments in the LOG file.

```
0 : only results in LOG
1 : description of important variables, choices of functions, ...
2 : everything (useful to locate a potential problem)
```

[Run]RemoveLOG : defines if the LOG file is removed after normal completion of the program. If **[Run]RemoveLOG** =yes, then it is deleted if no problem occurred.

[Run]Save : defines what has to be saved. As many options as wished.
Ex: save= atmosphere, raytrace,source, transmittance

[Run]StopAfter : indicates when ASIMUT will stop. This option is useful for example to extract the spectroscopic parameters (ExtractLP).

[Run]StopAfterError: When a list of several .INP is specified in **[List]**, stops after the first error encountered if =yes; continues to the next .INP file if = no.

[Run]SaveResults : defines the type of files saved (Ascii or/and Matlab), both options can be selected.

[Run]UseMatlab : defines if Matlab will be used.

[Run]Plot : indicates what will be plotted. If something is selected a XXX file is created containing the plots of the simulation or fit results.

[Run]PlotVisible : indicates if the plots are visible or not. When =0, the plots are done, but the focus is never set to their windows allowing further normal use of the machine while ASIMUT is running.

[Run]NormalizeSpectrum: indicates if the experimental spectrum has to be normalized. Normalization wrt the mean of the signal (same as SFIT).

The other options are related on how the Rodgers algorithm is applied. They will be described later.

[OnionPeeling]OnionPeeling: if =1, the onion peeling method will be used.

[Directories] : indicate the FULL directory where different quantities will be found. If these directories are mentioned, file names in the following .INP files are defined relative to these directories. If not given, file names in INP must contain full path information.

dirInput	Dir where the *.INP files are located
dirSpectra	Dir where spectra are located
dirInstrument	Dir where instrumental functions are located
dirAtmosphere	Dir where Atmosphere data are located
dirZPT	Dir where ZPT files are located
dirPlanet	Dir where planetary files are loaded (Emissivity, topography)
dirSolar	Dir where solar data are located
dirHitran	Dir where general Hitran/Geisa files are located
dirLP	Dir where local .mod/hitran/geisa files are located
dirApriori	Dir where a priori (Sa, TikonovFile) data are located
dirAerosol	Dir where the data relative to the aerosols are located
dirLidort	Dir containing a (V)LIDORT input flag file and the SPHER/T-MATRIX aerosol input file
dirMatlab	Dir where the macros for plotting with Matlab are located. Normally, this directory should also be added to the PATH variable in Matlab. Option exists in the case the user has not the rights to change this path (see Section 2.4 Troubleshooting).

[RadiativeCode] Code: Type of Radiative Transfer calculation: Asimut, Lidort. Two other options (Vlidort and VlidortF90 exist but they have not been tested fully: it is thus highly recommended NOT TO USE them).

[List]: gives the list of the INP files that will be processed using the same options defined in this ASI file.

2.2.7 Description of the output files

In the following, we will use FileASI and FileOUT to represent the base names of the files .ASI and the output files respectively. FileASI is constructed on the name of the file .ASI without its extension. The FileOUT radical is by default based on the base of the name of the file .INP, however if a *[Set]caption* is mentioned, it will define the ‘FileOUT’ radical.

For example: the files mars_nadir_IR.asimut and nadir_IR_CH4_test.inp files will be associated to FileASI=mars_nadir_IR and FileOUT =nadir_IR_CH4_test. If *[Set]caption=Nadir_Test2* is present inside the nadir_IR_CH4_test.inp file, then FileOUT=Nadir_Test2.

2.2.7.1 LOG file

A Log file is created at each run, with its name = FileASI_yymmdd.LOG. It overwrites any file having the same name.

This file contains information on the run itself: which parameters were considered (default values or user given), which options were selected, a rough indication of the calls to subroutines, indication of timing, a summary of the results, etc.

The level of details in the file is indicated by the parameter *[Set]Verbose*.

If *[Run]RemoveLog* is set to ‘Remove’ (or ‘yes’), then the .LOG file will be removed IF the programs ends without any problem.

If an error occurs, a series of warnings and explanations will be produced and written in the .LOG file. To find the origin of the problem, search the first warning. A short description of the error, the file (along with the function and line number) in which it happens. This information can be used to understand the origin of the problem (file not given, file not in the expected format, value for requested parameter not given, etc). In case of problems for which the solution is not trivial, do not hesitate to contact IASB-BIRA (see Annex 7).

2.2.7.2 General files in /RESULTS

Several files are always produced and are saved in the /RESULTS directory. They contain general information on the run. They can be saved in ASCII or in MATLAB format.

MATLAB File - FileOUT.mat

The MAT file contains 1 matrix structure (‘s’), one element for each pass or each spectrum defined in file .INP.

Each element is composed of a series of sub-structures or elements:

s =

nbfen:	16
fen:	[1x16 struct]
geo:	[1x16 struct]
nbspecies:	2
species:	[1x2 struct]
fit:	[1x1 struct]
den:	[1x1 struct]
vmr:	[1x1 struct]

where:

nbfen	is the number of Fenetre simulated or retrieved for this pass or spectrum
fen	is a matrix structure defining the Fenetre

geo	is a matrix structure defining the geometry for each of the Fenetre
nbspecies	is the number of species (retrieved + non-retrieved)
species	is a matrix structure defining the vertical profiles of the species included (retrieved + non-retrieved)
fit	is a structure defining the fit
den	is a structure defining the vertical densities (retrieved + non-retrieved)
vmr	is a structure defining the vertical mixing ratios (retrieved + non-retrieved)

Each element of `s(i).fen` contains

Ts	Brightness temperature (result of fit) (K)
Tsap	Brightness temperature (a priori value) (K)
Tserr	Error on b Brightness temperature (K)
B	4 parameters defining the background (result of fit) [background= $B(1)+B(2)*(nu-B(4))+ B(3)*(nu-B(4))^2$]
Bap	idem a priori values
Berr	idem error on fitted values (except B(4))
nu	Wavenumbers
obs	Observed spectrum
clc	Simulated spectrum
dif	(Observed-Simulated) spectrum

Each element of `s(i).geo` contains

Hobs	Observer altitude (km)
Hend	End altitude of light path (km)
Hmin	Tangent height (km)
type	0=HORIZ,1=NADIR,2=OFFAXIS,3=LIMBANG,4=LIMBTG,5=TOSPACE,6=GENERAL
sza	Solar zenith angle – Angle at HObs
beta	Angle at the Planet
phi	Angle at Hend
Re	Planet radius
lon	Longitude
lat	Latitude

Each element of `s(i).species` contains

name	Name of the species
z	Altitudes used for fitting the species
value	Fitted values (around 1)
err	Error on fitted values
den	Fitted densities
vmr	Fitted vmr

Each element of `s(i).fit` contains

n	Number of fitted parameters
---	-----------------------------

m:	Number of spectral points
A:	Averaging kernels
Sm:	Smoothing error
Sn:	Experimental error
dof:	Degree of freedom
rms:	Rms if fit

Each element of `s(i).den` contains

names:	Names of species
z:	Altitudes
ap:	A priori density values
rt:	Retrieved density values (= a priori values if not fitted)
er:	Errors on retrieved density values (= 0.0 values if not fitted)

Each element of `s(i).vmr` contains

names:	Names of species
z:	Altitudes
ap:	A priori vmr values
rt:	Retrieved vmr values (= a priori values if not fitted)
er:	Errors on vmr density values (= 0.0 values if not fitted)

ASCII File - FileOUT.out

The ASCII file contains all data read or computed during the run and corresponding to the parameters from file .INP. The file is divided into BLOCKs and GROUPs. For example all data concerning one Pass is saved into one BLOCK (called (Pass_n, where n is the Pass number, usually 1). Different GROUPs are defined all containing a series of Parameters (PARAM), data axis (AXIS) and data (DATA).

Matlab macros are provided (in @blockreader and @binput) to import easily the data contained in this file into Matlab, or to perform searches on selected variables.

Plot File - FileOUT.ps

If plots were requested, a single PostScript file is created containing different plots (radiance, transmittance, averaging kernels, etc).

2.2.7.3 Specific files in /SAVE

Files can be saved in ASCII or in MATLAB format. All files are created with a header indicating the date and time of creation and the content and units of each column.

Radiances and transmittances (see also Annex 4)

Depending on the *[SPn]DataType* flag, the following data will be saved in the FileINP_rad_forw.dat (and FileINP_rad_final.dat):

If [SPn]DataType = Radiance

WaveNumber	Simulated_Radiance	Observed_Radiance	Io_Radiance
Solar_Radiance	Source_Radiance		

If [SPn]DataType = Transmittance

WaveNumber	Simulated_Transmittance	Observed_Transmittance	
Simulated_Radiance	Observed_Radiance	Io_Radiance	Solar_Radiance
Source_Radiance			

The wavenumber column will be expressed in cm^{-1} (default) or nm depending on the *[SPn]DataXUnit* flag. All radiances are expressed in $(\text{W} \cdot \text{cm}^{-2} \cdot \text{sr}^{-1})$.

If no Spectrum is provided by the user, the columns ‘Observed_Radiance’ and/or ‘Observed_Transmittance’ disappear.

Io_Radiance = radiance used for the calculation of the Transmittance. It depends on the geometry:

- for solar occultation : SUN radiance
- for NADIR :
 - if no Sun reflection : Surface emission;
 - if Sun reflection included :
 - = $\text{SUN_radiance} \times \text{SolarAngle} / \pi$.

Solar_Radiance = radiance of SUN, if SUN is included in the RT calculation.

Source_Radiance = radiance of the Source indicated in the *[SPn]* section:

- For solar occultation = SUN radiance
- for NADIR : always Surface emission (including emissivity).

The definition of **Simulated_Transmittance** depends on the geometry:

- For solar occultation = $\text{Simulated_Radiance} / \text{SUN_radiance}$
- for NADIR :
 - if no Sun reflection : $\text{Simulated_Radiance} / \text{Surface_radiance}$
 - if Sun reflection : $\text{Simulated_Radiance} \times \pi / \text{SUN_irradiance} = \text{Radiance Factor}$

2.2.8 Examples Provided

Different examples are provided with the code. A general one (in the EXAMPLE directory) illustrating different options of the forward model (with and without LIDORT). A second series of examples (in the EXAMPLE_SOIR directory) is more specific to the SOIR instrument) is also given and illustrate the simulation and retrieval of SOIR spectra. In each case, the full set of ancillary data is provided, as well as output files (LOG, OUT and data files) to check that the runs are done correctly.

2.2.8.1 Extracting Line parameters

Uses : **mars_nadir_IR_lp.asi**, calling **nadir_IR_CH4_lp.inp**.

Shows how to call ASIMUT-ALVL to extract the spectroscopic line parameters for a specific run.

ASIMUT will stops after the extraction of the spectroscopic files which will be used by Example 2.

2.2.8.2 Nadir IR spectra of Mars atmosphere – no aerosols

Uses : **mars_nadir_IR.asi**, calling **nadir_IR_CH4_noAerosol.inp**.

This example shows how to call ASIMUT-ALVL to simulate an IR spectrum of the Mars atmosphere under nadir viewing. No aerosol are included in the run, only CO₂, CH₄ and H₂O are included as Line-by-Line molecules (LP).

2.2.8.3 Nadir IR spectra of Mars atmosphere – with aerosols

Uses : **mars_nadir_IR.asi**, calling **nadir_IR_CH4_AerosolMode1.inp** or **nadir_IR_CH4_AerosolMode2.inp**.

Also uses **mars_nadir_IR_Lidort.asi**, when running under UNIX with the ASILID version (link between ASIMUT and LIDORT).

This example shows how to call ASIMUT-ALVL to simulate an IR spectrum of the Mars atmosphere under nadir viewing. CO₂, CH₄ and H₂O are included as Line-by-Line molecules (LP), as well as two types of aerosols, illustrating Angström type aerosols and LidortG type.

The examples have been run under ASIMUT (windows) and under UNIX with the ASIMUT and LIDORT options. When using the ASIMUT option, only the extinction of the aerosols are included (no scattering effect in ASIMUT). With the type (Mode2:LidortG) aerosol, scattering due to these aerosols are also included when using the LIDORT option. For each INP file, 3 sets of output files are created: one using ASIMUT under Windows, one using ASILID with the [Run]Code=ASIMUT optin (files ending by ‘_A’), and one using ASILID with the [Run]Code=LIDORT optin (files ending by ‘_L’).

The plot (RESULTS\ Comparison_Aerosols.fig and Comparison_Aerosols.png) illustrates the comparison of the radiances obtained in the different runs.

2.2.8.4 Analysis of a SOIR solar occultation observation

Different test cases are given: simulation of an AOTF like spectrum, reading of a PDS SOIR spectrum, reading of a CSV SOIR spectrum, analysis of a series of individual spectra, combined analysis of a series of spectra (onion peeling).

Venus_149.asi + 341_149_1_onlySimulation.inp

No observed spectrum is provided, all information on the instrument and observation) has to be specified in the .INP file.

Venus_149.asi + 341_149_1_singlePDSspectrum.inp

A SOIR PDS type of file is provided, all information on the instrument and the observation are read from the header. This file format is identical to the one used in the SOIR PSA (ESA) archive.

Venus_149.asi + 341_149_1_singleCSVspectrum.inp

A SOIR CSV type of file is provided, all information on the instrument and the observation are read from the header. The MATLAB macro *ConvertPDS_ASCII.m* used to convert PDS file to CSV and to create some additional files (such as the LST file, giving the list of all files converted) is also provided in the ASIMUT\MATLAB subdirectory.

Venus_149.asi + 341_149_1_severalCSVspectra.inp

Individual analysis of the spectra listed in the *20070820_I01_149_Sub.Lst* list. This list has been created from the full list of spectra (*20070820_I01_149.Lst*), by selecting one spectrum every 5 km. All spectra are analysed sequentially and independently.

Venus_149.asi + 341_149_1_severalCSVspectraTogether.inp

Same as previous but all spectra are analysed simultaneously (only ONE fit of the CO2 profile considering the 6 spectra all together).

Venus_149_OnionPeeling.asi + 341_149_1_severalCSVspectra_onionPeeling.inp

Same as previous, but all spectra are analysed using the Onion Peeling method. Only ONE CO2 profile is reconstructed from the analysis of the 6 spectra.

2.2.9 On-line help

A web based online help is also provided. It contains general information on the ASIMUT-ALVL project, but also a full description of all the .h files, all structures, variables, functions, ... being part of the code.

The *ASIMUT\Documentation\ASIMUT_Documentation.html* contains a (WINDOWS) link to the index of the web based online help, which is *ASIMUT\Documentation\html\index.html*.

Remark : under WINDOWS, the link contains the FULL path to the index.html file, please do not forget to change the full path in the Properties of the *ASIMUT\Documentation\ASIMUT_Documentation.html* file.

Under UNIX, start the online page by selecting the index file (*ASIMUT\Documentation\html\index.html*).

The on-line help is also accessible on the Planetary Aeronomy webpage:
http://planetary.aeronomie.be/en/asimut_documentation/html/index.html

2.3 How to

In this section we will give some advice how to perform specific operations. This covers for example, ‘How to start ASIMUT ?’, ‘How to extract spectroscopic data ?’, etc., but also issues on how to add specific code inside ASIMUT. These are the most asked questions about how ASIMUT runs and works.

2.3.1 How to start ASIMUT ?

Under UNIX:

Change directory to be where the ASILID executable is located (or provide the full path to the ASILID executable in the following line).

`./ASILID fullPath/file.ASI`

Under Windows, within the VISUAL environment:

Select the mode in which you want to run ASIMUT : Debug or Release

Change the ‘Properties’ → ‘Configuration Properties’ → ‘Debugging’ : set the ‘Command arguments’ to `fullPath\file.ASI`

Click on the ‘Green arrow’ or select ‘Start debugging’

Under Windows, outside the VISUAL environment:

Open a ‘cmd’ window;

Change directory to where the executable is located;

Type : `Asimut fullPath\file.ASI`

2.3.2 How to extract spectroscopic (LP) data from literature database ?

In the ASI file , set:

```
[Run] save=lp  
[Run] StopAfter=ExtractLP
```

Under [List], specify one INP file representative of what will be done (same number of FEN and the limits that will be used). In the INP file do not forget to remove (or place in comments using % at the start of the line) the lines defining the name of the files in the [xxxLP] sections.

Run Asimut: the MOD files will be created with all the information needed for this specific call to Asimut.

The two files (*mars_nadir_IR_lp.asi* and *nadir_IR_CH4_test_lp.inp*) in the *Example* directory give you an example of such calling.

2.3.3 How to limit the size of the .LOG file ?

Set [Run] verbose to a lower value.

2.3.4 How to add a new Instrument ?

To include a new instrument is easy but requires the modification of several files. First of all a specific NEWINSTRUMENT.c (and accompanying NEWINSTRUMENT.h) file has to be built, following the template proposed in Annex 6: Template of file for a new instrument). The content of this file will depend on what type of input files are read. Several options are available:

- (1) Only one file containing one observation (spectrum);

- (2) A list containing a series of filenames that have to be treated in the same way (same options of INP files), but each file being analysed independently; (option also used with the Onion Peeling method)
- (3) A list containing a series of filenames that have to be treated in the same way (same options of INP files), but all files being analysed simultaneously (Occultation type of observation).

To follow the existing ASIMUT structures, those files should be saved in the *ASIMUT/ASIMUT/Spectra* directory.

If you are working under SVN:

- (a) Add the new files in the SVN structure: in your browser (under Windows), right-click on the file to add, select “SVN → Add”
- (b) Change the properties of the files: right-click on the file, select ”SVN → Properties”. A dialog box appears, in which you add the following: “svn:keywords LastChangedDate LastChangedRevision LastChangedBy”

This will allow the automatic change of the files headers (see Templates) when the files are modified.

Be sure that those headers are present:

```
/*! \file NEWINSTRUMENT.h
   \brief Functions relative to NEWINSTRUMENT

   Created : 07/03/2006
   $Rev::: 257      $
   Last modified : $Date::: 2009-10-01 16:31:12#$
   by $Author::: xxx                         $

*/
```

Specify the new .h file to be included in the ASIMUT project: add the following line in the *ListAllh.h* file (in *ASIMUT\ASIMUT\common*)

```
#include "../Spectra/NEWINSTRUMENT.h"
```

The second place to change is inside the existing *User_Choice.c* (and *User_Choice.h*). Each modification will be explained in the following for all three types of NEWINSTRUMENT files.

First, add the new instrument name in the existing list. In *User_Choice.h*, add the name (all capitals) at the end of the *enum* field (lines 44 sq) defining ‘*SpecType*’.

```
/*! Type of Spectrum File */
enum SpecType { FILE_NONE, FILE_OPUS, FILE_ASCII, FILE_SFIT, FILE_CSV,
    FILE_OPUSINFOCSV, FILE_IMGCSV, FILE_IASICSV,
    FILE_SOIRCSV, FILE_SOIRPDS3, FILE_ACE};
```

Becomes:

```
/*! Type of Spectrum File */
enum SpecType { FILE_NONE, FILE_OPUS, FILE_ASCII, FILE_SFIT, FILE_CSV,
    FILE_OPUSINFOCSV, FILE_IMGCSV, FILE_IASICSV,
    FILE_SOIRCSV, FILE_SOIRPDS3, FILE_ACE, FILE_NEWINST};
```

For option (1), only the *Read_NEWINST (...)* function is needed in the *NEWINSTRUMENT.c* file.

Add the following lines inside User_Choice.c in the *ReadSpectrum_Choice(...)* function(for example, just before the ‘else’):

```
else if ( (SameString(ChoiceStr, "NEWINST")) && ((*IsList)==NO_LIST) )
{
    *type=FILE_NEWINST;
    ReadFct=(Read_NEWINST);
}
```

For option (2), the *Read_NEWINST (...)*, *GetNbSpec_NEWINST_LIST(...)*, *GetNameSpec_NEWINST_LIST(...)*, and *ModifySpec_NEWINST_LIST(...)* functions are needed in the NEWINSTRUMENT.c file.

Add the following lines inside User_Choice.c in the *ReadSpectrum_Choice(...)* function(for example, just before the ‘else’):

```
else if ( (SameString(ChoiceStr, "NEWINST")) && ((*IsList)==LIST) )
{
    *type=FILE_NEWINST;
    ReadFct=(Read_NEWINST);
    *GetNbSpectra =(GetNbSpec_NEWINST_LIST);
    *GetNameSpectrum=(GetNameSpec_NEWINST_LIST);
    *ModifySpectrum=(ModifySpec_NEWINST_LIST);
}
```

For option (3), the *Read_NEWINST (...)*, *GetNbSpec_NEWINST_OCC(...)*, *GetNameSpec_NEWINST_LIST(...)*, and *ModifySpec_NEWINST_LIST(...)* functions are needed in the NEWINSTRUMENT.c file.

Add the following lines inside User_Choice.c in the *ReadSpectrum_Choice(...)* function(for example, just before the ‘else’):

```
else if (SameString(ChoiceStr, "NEWINST_OCC"))
{
    *IsList=OCC;
    *type=FILE_NEWINST;
    ReadFct=(Read_NEWINST);
    *GetNbSpectra =(GetNbSpec_NEWINST_OCC);
    *GetNameSpectrum=(GetNameSpec_NEWINST_LIST);
    *ModifySpectrum=(ModifySpec_NEWINST_LIST);
}
```

2.4 Troubleshooting

2.4.1 WIN: Debugging information for "project.exe" cannot be found or does not match. Binary was not built with debug information.

To enable debugging:

1. Go to Project → Project Properties
2. On the left expand "Configuration Properties"
3. On the left, expand "Linker"
4. On the left, select "Debugging"
5. On the right, change "Generate Debug Info" to "Yes"
6. Click ok
7. Set your breakpoints
8. Rebuild your application

2.4.2 WIN: No compilation: "Fatal error C1033: cannot open program database "

Be sure that you are Administrator on your machine, or that you have started VisualC with the option 'Run as Administrator' (right-click)

2.4.3 WIN/UNIX: The figures from MATLAB are created but are empty

This error is also accompanied with the following warnings during the execution of the program:

*??? Undefined function or method 'PlotZPT' for input arguments of type 'double'.
??? Undefined function or method 'AddString' for input arguments of type 'double'.
??? Undefined function or method 'PlotSpecSimul' for input arguments of type 'double'.*

Check that ASIMUT/MATLAB directory is included into the Path of MATLAB and saved. To do that: select the 'File' menu, then 'Set Path...', click on 'Add with Subfolders' and then select 'ASIMUT/MATLAB'. Then 'Save'. Windows: Note that you have to be 'Administrator' on your machine to be authorized to save. If you are not logged as Administrator, you can right-click on MATLAB and select the 'Run as Administrator' option. You will be asked to enter your Administrator login.

Another (easier) solution is to mention the ASIMUT/MATLAB directory under the *[directories]dirMatlab* in the .ASI file:

```
[Directories]
dirMatlab= C:\MyProjects\Asimut\Matlab
```

2.4.4 WIN/UNIX: Problem with Files

If you encounter a problem with files that apparently are in line with the input expected to be read from them, check

- (1) That they are not open by another program. If it is the case, close that program and start ASIMUT again;
- (2) Check that you have the right access permission to the files and directories, including 'write' permission for the output files;

- (3) Check the endings of the line: under Windows, it must be CR-LF ('r\n'), under UNIX only CR ('\n');
- (4) Under UNIX, check that the name of the directories and files contain only '/' (no '\' are allowed). This is not a problem under WINDOWS.

3. Radiative Transfer simulation

The ASIMUT program simulates the atmospheric transmittance and radiance in cases where scattering can be neglected and under conditions corresponding to local thermodynamic equilibrium. The model assumes a one-dimensional atmosphere, with all parameters varying only with the vertical variable, the altitude. ASIMUT can be used to simulate measurements recorded under various geometries (nadir, off-axis nadir, down and up looking, limb looking) with the instrument located at ground level, inside the atmosphere, or in space.

The equation describing the radiative transfer through the atmosphere can be written as:

$$I(\nu) = I_0(\nu)e^{-\tau(\nu,0,z)} + \int_0^z B(\nu, T(s))\alpha(\nu, s)e^{-\tau(\nu,0,s)}ds \quad (3.1)$$

where I_0 represents the light intensity at the starting point of the ray path situated at the distance z from the observer, $\alpha(\nu, s)$ is the absorption coefficient, $B(\nu, T)$ is the Planck function, and

$$\tau(\nu, s_1, s_2) = \int_{s_1}^{s_2} \alpha(\nu, s)ds \quad (3.2)$$

is the optical depth along the path between the points s_1 and s_2 . The integration in Eq.(3.1) occurs along the ray path, which must be known beforehand.

For **nadir looking** viewing mode, I_0 is composed of several terms : (a) the radiation emitted by the ground, simulated by a blackbody function whose temperature might be fitted during the retrieval procedure; (b) the reflection of the thermal emission sources originating from the different layers of the atmosphere (specular reflection); and (c) the reflected solar radiation attenuated through the atmosphere.

For measurements looking directly at the sun, I_0 can be either simulated by a blackbody function or read from an external file, allowing for example the use of the high resolution Kurucz atlas [16] or solar measurements [17].

The simulation of an atmospheric spectrum can be separated into three distinctive components. It first of all requires the determination of the ray path through the atmosphere and the computation of the temperature, pressure and molecular densities along this path. Then, using this information and the line parameters or absorption cross sections relative to each absorbing species, the optical depths corresponding to each layer are determined. Finally, the radiative transfer itself must be solved. The three steps will be described in more details in the following sections.

3.1 Ray tracing

The determination of the radiation path through the atmosphere, i.e. the path followed by the radiation reaching the instrument, requires that the Earth's curvature and refraction be taken into account. The model is based on the ray-tracing program FSCATM [18]. FSCATM has been improved when being incorporated into the line-by-line program LBLRTM [19, 20] and later when included into the retrieval program SFIT[21]. However some modifications were included in the ASIMUT implementation, essentially dealing with the determination of the index of refraction.

3.1.1 Index of refraction

Ray-tracing performed in FSCATM uses the equation of Edlen [22] including a correction for the presence of water vapour. As discussed in Ciddor [23] and in Tomasi et al.[24], new measurements of dispersion and of absolute refractive index have since been made. Ciddor [23] developed a new set of equations based on the most recent equations for air density and air dispersion in the visible and near infrared regions. The moist air refractivity $n(\nu, P, T, e, C)$ at nonstandard temperature T and pressure P conditions and taking into account the presence of water vapour (with partial pressure e) and of CO_2 (with volume concentration C) is calculated as the sum of a first term giving the refractivity of dry air ($e=0$) containing CO_2 multiplied by its density normalized to that of standard dry air conditions ($T_s=288.15\text{K}$ and $P_s=1013.25\text{ hPa}$) , and a second term giving the refractivity of pure water vapour $n_{H_2O}(\nu, T^*, e^*)$ multiplied by its density normalized to that defined at standard conditions ($T^* = 293.15\text{ K}$ and $e^* = 1333\text{ Pa}$):

$$n(\nu, P, T, e, C) - 1 = \frac{\rho(P, T, 0, C)}{\rho(P_s, T_s, 0, C)} [n(\nu, P_s, T_s, 0, C) - 1] + \frac{\rho(e, T, e, 0)}{\rho(e^*, T^*, e^*, 0)} [n_{H_2O}(\nu, T^*, e^*) - 1] \quad (3.3)$$

The detailed description of how to determine all the parameters appearing in Eq.(3.3) can be found in Ciddor [23] and Tomasi et al.[24]. As Ciddor's equation gives somewhat better results on a broader range of wavelengths and under extreme environmental conditions of temperature, pressure and humidity, it has been adopted by the International Association of Geodesy as the standard equation for calculating the index of refraction. In fact the Edlen equation includes approximations for the water vapour term that are only accurate at 20 °C, and therefore could be in error at high temperature and humidity.

The default subroutine used to determine the index of refractivity is thus the Ciddor one, at least for the Earth. This code is not at all adapted for other planets' atmospheres. Different other possibilities have thus been included:

Lblrmt [19, 20] : same expression as used in the LBLRTM code, which is expressed in terms of λ , λ^2 , and λ^4 , including a small correction for the water vapour;

Sneep [25]: determination of the total refractive index of 'air', by considering the individual contribution of CO_2 , CO , N_2 , O_2 , H_2O , and Argon. The general expression of this function makes it applicable for all planets' atmospheres;

Magellan data: for Venus, the index of refraction can be determined using the measurements performed by Magellan;

Mars: parameterization using only the altitude as variable.

3.1.1.1 [Planet] Options

These options can be selected in the .INP file by imposing the *[Planet]RefractiveIndex* parameter. General description of the planet characteristics are to be supplied under *[Planet]* (*Planet*, *PlanetRadius* (km), *DistToSun* (AU), *Topography*). Defaults values are taken into account depending on the choice of the *[Planet]Planet* parameter (Earth, Mars or Venus). Other planets or bodies can be simulated but then respective values of the parameters must be supplied.

3.1.2 Atmospheres

The user may either choose one of the built-in atmospheres (US Standard Atmospheres [26], MIPAS Model Atmospheres [27]) or propose his own volume mixing ratio (vmr) vertical profiles, as well as the vertical profiles for temperature and pressure.

3.1.2.1 [AtmosphericModels] options

[AtmosphericModels]Model : The user can specify one of the hard-coded atmospheres for the Earth or a 'user defined' file (*Model=0*). There are also hard-coded models for the atmospheres of Venus (VIRA day model) and Mars (typical atmosphere, South Winter), but those do not cover a wide series of species (for ex., only CO₂ and N₂ in the case of Venus). For these planets, the user is advised to give a user defined atmosphere. For Earth 6 different atmospheres have been hard-coded. They correspond to the Tropical, Mid-latitude summer, Mid-latitude winter, Subarctic summer, Subarctic winter of the US standard Model, as well as the US Standard 1976 Model (*Model=1* to *6*, respectively). The **[AtmosphericModels]Default** correspond to the hard-coded atmosphere that is used when a species is not found in the atmosphere provided by the user but is required (for example, to calculate the index of refraction by the Snell method, CO₂, CO, N₂, O₂, H₂O are needed, if not present in the user-defined atmosphere, values will be taken from the *Default* hard-coded one).

[AtmosphericModels]atmFile - **[AtmosphericModels]atmFileType** : indicate the name and type of the file containing the atmosphere. Several types are supported:

- *Col*: ascii columns with VMR values (expressed in ppm), altitudes are expressed in km, the first line must start with %# and contain the label of each of the columns, lines starting with % are comments;

Example : venus_day_ppm.dat
% Z(km) NT(cm-3) CO2(ppmv) N2(ppmv) CO(ppmv) O3P(ppmv)
%# Z N CO2 N2 CO O3
6.0000e+001 4.7200e+018 9.6500e+005 4.6200e+001 5.1000e+001 2.1200e-007
6.1000e+001 3.9700e+018 9.6500e+005 4.7000e+001 4.1200e+001 3.0300e-006
...
• *Sfit*: SFIT type of file, provided by the REFMOD program);
• *Mipas*: Mipas format file.

[AtmosphericModels]zptType - **[AtmosphericModels]zptFile** : indicate the type and name of ZPT file. This file contains information on the pressure and temperature structures versus altitude. Several types are supported:

- *Ascii*: ascii file with 3 columns (z[km] p[mb] t[K]);
- *Ncep*: NCEP file format (used by SFIT2, defined on 41 levels);
- *Ecmwf*: ECMWF file format (only the 3 first columns, z, p, and t are read);
- *EcmwfRH*: ECMWF file format (z, p, t and Relative Humidity are read). Relative Humidity is then converted to H₂O vmr and imported in the Atmosphere;
- *Std*: z p t are read in the **[AtmosphericModels]atmFile** (z [km] p[mb] t[K]);

Example : venus_day_ppm.dat
% Z(km) T(K) P(mb) NT(cm-3) CO2(ppmv) N2(ppmv) CO(ppmv) O3P(ppmv)
%# Z T P N CO2 N2 CO O3
60.0 2.7278e+002 1.780e+002 4.72e+018 9.65e+005 4.62e+001 5.10e+001 2.12e-007
61.0 2.6894e+002 1.470e+002 3.97e+018 9.65e+005 4.70e+001 4.12e+001 3.03e-006

[AtmosphericModels]Density : indicates how the water contribution is removed from the determination of the total air density.

[AtmosphericModels]Space : definition of the altitude (km) of where space begins. No calculations of optical depths or contribution to the radiative budget are performed above this limit. If not given, the value is the highest altitude in the **[AtmosphericModels]atmFile**.

3.1.3 Geometry of observation

Several geometries are supported (nadir, off-axis nadir, from ground to space, from space to ground, solar occultation). The user further needs to give some of the following parameters to fully describe the ray path: the position of the observer (H_{obs}), the looking direction or zenithal angle (ϕ), the tangent height for limb measurements (H_{tg}), and the altitude of the end point of the light path (H_s). With the help of those parameters, any configuration may be simulated (Figure 3). The zenithal angle can be specified either as the apparent or astronomical angle. In the latter case, an iterative procedure is applied to determine the apparent angle.

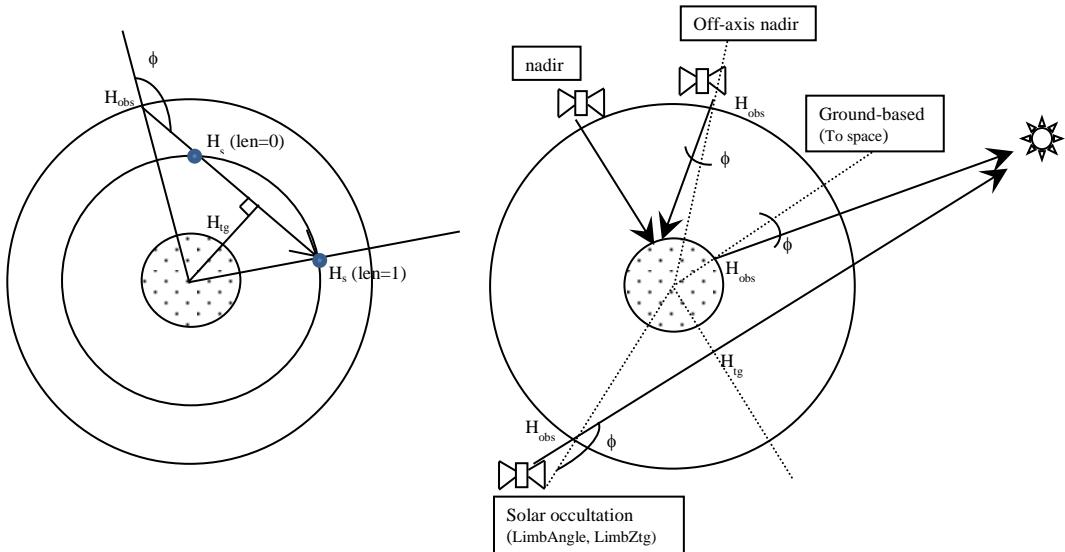


Figure 3: Several geometries are possible. H_{obs} is the altitude of the observer, ϕ is the zenithal angle, H_{tg} is the tangent height, and H_s is the end point of the ray path.

The user specifies a layer structure on which the optical densities and radiative transfer calculations will be performed. The ray-tracing calculations are carried out on a finer grid (with a 200 m step) and the final results, i.e. the effective temperature and pressure for each layer as well as the air density, are obtained using the Curtis-Gordon approximations [28].

3.1.3.1 [Geometry] options

All parameters describing the Geometry of the observation are indicated under the **[SPn]** section.

[SPn]Geometry : indicates the type of geometry of the observation: horizontal path, nadir looking, nadir off-axis, limb observation (*LimbAngle*: when the angle ϕ , or *LimbZtg*, when the tangent height is specified), or looking to space.

[SPn]Refraction : indicates if refraction of the atmosphere has to be considered for the Ray tracing calculations. For example, not necessary on Mars.

[SPn]HObs : indicates the altitude of Observer (km). It MUST be given, except for 'nadir' and 'offaxis'. If not given for 'nadir' and 'offaxis', the space limit (*[AtmosphericModels]Space*) taken. See also on the specific definition of the Spectra files for some instruments, since this parameter could be given in their header (this is the case for SOIR spectra, for example).

[SPn]HEnd : indicates the altitude of the ending point of the LOS (km). It represent the tangent height when *[SPn]Geometry* = *LimbZtg*. See also on the specific definition of the Spectra files for some instruments, since this parameter could be given in their header (this is the case for SOIR spectra, for example).

[SPn]Angle : indicates the angle ϕ (degrees) defining the LOS. See Figure 3 for the different definition of the Angle in function of the geometry type. See also on the specific definition of the Spectra files for some instruments, since this parameter could be given in their header (this is the case for SOIR and IASI spectra, for example).

[SPn]AngleType : indicates if the angle ϕ is apparent or astronomical.

[SPn]len : in some case (see Figure 3) two LOS can be defined by specifying only the Hobs and HEnd altitudes. If *[SPn]len* = 1, it will consider the longer path, if =0, the shorter one.

[SPn]Range : length (km) of LOS when *[SPn]Geometry* = *Horizontal*.

[SPn]Sun_sza : Solar zenith angle (degree) w.r.t. the topocentric reference frame.

[SPn]VZA : Viewing zenithal angle (degree) w.r.t. the topocentric reference frame.

[SPn]SolAZ : Solar azimuth angle (degree) w.r.t. the topocentric reference frame.

[SPn]ObsAZ : Viewing azimuth angle (degree) w.r.t. the topocentric reference frame.

3.2 Optical depths

The absorption coefficient $\alpha(v, z)$ is in general the sum of several contributions: single absorption lines of different gaseous species, absorption cross sections and continua contribution, and the Rayleigh scattering term.

$$\alpha(v, z) = \sum_i n_i \sum_j S_{ij}(T) \phi(v, v_{ij}, T, P, p_i) + \sum_c n_c \sigma_c(v, T, P, p_c) + \alpha_{Rayleigh}(v, z) \quad (3.4)$$

where P and T are the pressure and temperature at altitude z , n_i and p_i the density and the partial pressure of species i at altitude z . The first term of the right hand side of the equation represents the absorption due to single absorption lines. The sum on j represents the sum on all transitions v_{ij} participating in the absorption of light. The second term represents the broadband absorption of light due to molecules which do not present resolved absorption structures. And the last term is, not *in se* an absorption term, as it corresponds to the Rayleigh scattering, but it is nevertheless taken into account at this point, as it is similar to a diminution of light through the atmosphere. These terms will be discussed in more details in the following.

3.2.1 Line-by-line calculation

The absorption coefficient k_{ij} ($\text{cm}^2 \text{ molecule}^{-1}$) for a particular line j of species i is given as:

$$k_{ij}(v, T, P, p_i) = S_{ij}(T) \times \phi(v, v_{ij}, T, P, p_i) \quad (3.5)$$

where $\phi(\dots)$ is the normalized line shape

$$\int_{\text{line}} \phi(v') dv' = 1 \quad (3.6)$$

The intensity $S_{ij}(T)$ exhibits a temperature dependence which can be described as:

$$S_{ij}(T) = S_{ij}(T_0) \frac{Q_i(T_0)}{Q_i(T)} e^{\left[\frac{c_2 E_j}{T_0} - \frac{1}{T} \right]} \frac{1 - e^{-\frac{c_2 v_{0j}}{T}}}{1 - e^{-\frac{c_2 v_{0j}}{T_0}}} \quad (3.7)$$

where c_2 is the second Planck constant hc/k_B (1.4387 cm K), with h the Planck constant, k_B the Boltzmann constant, and c the speed of light v_{0j} is the central wavenumber of the j^{th} transition, E_j is the energy of the lower state (cm^{-1}), and $S_{ij}(T_0)$ is the intensity at the reference temperature T_0 ($\text{cm}^{-1} \text{ molecule}^{-1} \text{ cm}^2$).

$Q(T)$ and $Q(T_0)$ are the total partition functions under local thermodynamic equilibrium conditions, at temperature T and T_0 , respectively. These functions describe the temperature dependence of the line intensity of the transition.

The values of these functions can be determined by ASIMUT by two means (see *[Molecules] PartitionFct* parameter):

- (1) (default) The latest TIPS table [29] based on HITRAN2008 has been implemented and is read by ASIMUT in function of the molecule, isotopologue and temperature. The values are given for temperatures ranging from 70 K to 500 K with a step of 1 K. For the moment, no interpolation is used, the value corresponding to the closest temperature is used.
- (2) The partition functions are approximated by:

$$Q(T) = a_0 + a_1 T + a_2 T^2 + a_3 T^3 \quad (3.8)$$

where a_0 , a_1 , a_2 , and a_3 are tabulated coefficients [30]

ASIMUT allows the user to select between different line profiles, the Voigt profile being the default one. Natural broadening of the lines, which occurs as a consequence of the uncertainty in photon energy, is small. Collision broadening is caused by secondary interaction with other gas molecules and atoms. Absorption lines under collisional regime are described by a Lorentzian function. Doppler broadening describes the statistical distribution of frequency shifts due to thermal motion. This effect is characterized by a Gaussian function and is most significant under low pressure conditions when the collision broadening effect is small.

The **Doppler** profile ϕ_D of a line centred on ν_{0j} wavenumber

$$\phi_D(\nu) = \sqrt{\frac{\ln 2}{\pi}} \frac{1}{\gamma_D} \exp\left(-\ln 2 \left(\frac{\nu - \nu_{0j}}{\gamma_D}\right)^2\right) \quad (3.9)$$

is characterized by its half width γ_D which depends on the molecular mass M of the absorbing species, as well as on the temperature

$$\gamma_D = \nu_0 \sqrt{\frac{2 \ln 2 k_B T}{M c^2}} \quad (3.10)$$

The half width γ_L of the **Lorentzian** profile ϕ_L

$$\phi_L(\nu) = \frac{1}{\pi} \frac{\gamma_L}{(\nu - \nu_{0j})^2 + \gamma_L^2} \quad (3.11)$$

is composed of a self-broadening contribution and of a foreign broadening contribution, which are pressure and temperature dependent:

$$\gamma_L = p_i \gamma_L^{self} + (P - p_i) \gamma_L^{foreign} \quad (3.12)$$

and

$$\gamma_L^{self} = \gamma_L^{self}(T_0) \left(\frac{T_0}{T}\right)^{n_{self}} \quad \gamma_L^{foreign} = \gamma_L^{foreign}(T_0) \left(\frac{T_0}{T}\right)^{n_{foreign}} \quad (3.13)$$

where p_i is the partial pressure of species i , P_0 and T_0 are the reference pressure and temperature and n_{self} and $n_{foreign}$ are the temperature coefficients of respectively the self and the foreign broadenings. The values of γ_L^{self} , $\gamma_L^{foreign}$ ($\text{cm}^{-1} \text{ atm}^{-1}$) and n_{self} , $n_{foreign}$ (unit less) are tabulated values in the databases. Pressure induced shifts are also taken into account in the determination of the line absorption. The central wavenumber is displaced by δ :

$$\delta = \delta_{self} p_i \left(\frac{T_0}{T}\right)^{n_{self}^s} + \delta_{foreign} (P - p_i) \left(\frac{T_0}{T}\right)^{n_{foreign}^s} \quad (3.14)$$

if self (δ_{self}) and foreign ($\delta_{foreign}$) shifts are considered. n_{self}^s and $n_{foreign}^s$ are the temperature coefficients of respectively the self and the foreign shifts. The shifts δ can be found in the databases as well but the temperature dependences are not available (yet) and has been taken as being identical to the values affecting the broadening coefficients (n_{self} and $n_{foreign}$).

When both broadening effects must be considered, the absorption line is best reproduced by a **Voigt** profile ϕ_V , which is the convolution of the Doppler and Lorentzian profiles:

$$\phi_V(\nu) = \sqrt{\frac{\ln 2}{\pi}} \frac{1}{\gamma_D} K(x, y) \quad (3.15)$$

where

$$K(x, y) = \frac{y}{\pi} \int_{-\infty}^{+\infty} \frac{e^{-t^2}}{y^2 + (x-t)^2} dt = \operatorname{Re} [W(x, y)] \quad (3.16)$$

in which

$$x = \sqrt{\ln 2} \frac{\nu - \nu_{0j}}{\gamma_D} \quad \text{and} \quad y = \sqrt{\ln 2} \frac{\gamma_L}{\gamma_D} \quad (3.17)$$

and $W(x, y)$ is the complex error function

$$W(x, y) = \frac{i}{\pi} \int_{-\infty}^{+\infty} \frac{e^{-t^2}}{x + iy - t} dt \quad (3.18)$$

Different methods to calculate Voigt profile are implemented : Pade approximation, Humlicek [31], or Faddevaa [32].

Intramolecular collisions have, as already explained, a broadening effect on the line shape. But they also perturb the translation motion of molecules, which implies a reduction of the broadening related to the Doppler effect. This effect is often referred to as the Dicke narrowing of the line shape. To model this effect, a further parameter has to be included and the line shape is modified. Two theoretical models of the effect, the soft and hard collision theories, lead to two slightly different line profiles. Under the soft collision hypothesis [33], the added parameter β introduced to describe the narrowing effect, is equal to $\beta^0 P$ where β^0 is the narrowing parameter ($\text{cm}^{-1} \text{atm}^{-1}$). This parameter should be compared to the dynamical friction β_{diff}^0 which is deduced from the mass diffusion coefficient D_{12} :

$$\beta_{\text{diff}}^0 = \frac{k_B T}{2\pi c m D_{12}} \quad (3.19)$$

The line shape is described by a **Galatry** function [33]:

$$\phi_G(\nu) = \frac{1}{\sqrt{\pi}} \operatorname{Re} \left[\frac{1}{\frac{1}{2z} + y - ix} M \left(1; 1 + \frac{1}{2z^2} + \frac{y - ix}{z}; \frac{1}{2z^2} \right) \right] \quad (3.20)$$

with

$$z = \sqrt{\ln 2} \frac{\beta}{\gamma_D} \quad (3.21)$$

and M is the confluent hypergeometric function

$$M(a, b, c) = 1 + \frac{a}{b} c + \frac{a(a+1)}{b(b+1)} \frac{c^2}{2!} + \frac{a(a+1)(a+2)}{b(b+1)(b+2)} \frac{c^3}{3!} + \dots \quad (3.22)$$

When the hard collision theory is used, the line shape is described by the **Rautian** profile [34]:

$$\phi_R(\nu) = \operatorname{Re} \left[\frac{W(x, y+z)}{1 - z\sqrt{\pi} W(x, y+z)} \right] \quad (3.23)$$

where $W(x,y)$ is the complex error function already defined in Eq. (3.23). Both the Galatry and Rautian profiles are implemented in ASIMUT.

In the case of the H_2O and CO_2 molecules, **sub and super-Lorentzian** line profiles have been observed [35, 36], which are characterized by a symmetric or asymmetric χ function, respectively:

$$\phi_\chi(\nu) = \chi(\nu) \times \phi_L(\nu, P, T) \quad (3.24)$$

This correction factor has been introduced to take into account the fact that far from the line centre, the line displays marked deviations from a Lorentzian behaviour. Typically, CO_2 displays a sub-Lorentzian behaviour (the opacity far from the line centre is less than that predicted by a Lorentzian profile) whereas H_2O shows a super-Lorentzian behaviour. The χ function is usually defined on large spectral intervals and must be provided by the user.

3.2.1.1 Grid and resolution used for the OD simulation

Sampling of the lines - The grid, on which the profile must be determined, must be fine enough so that the narrowest line be adequately represented. In the upper atmosphere, the Doppler width is the limiting factor for selecting an adequate sampling value. As Doppler width depends on temperature, it varies with altitude; the line profile must therefore be sampled with different steps as the altitude varies. ASIMUT determines for each layer, of temperature T and pressure P , the optimized sampling step as:

$$\Delta\nu(P, T) = \frac{1}{4} \sqrt{\alpha_D^2(T) + \alpha_L^2(P, T)} \quad (3.25)$$

with α_D the Doppler width obtained for a molecule of mass 20.0, and α_L the Lorentzian width obtained for a molecule characterized by no self-broadening, a foreign broadening of $0.04 \text{ cm}^{-1}/\text{atm}$, and a temperature coefficient of 0.5. However, for some particular temperature and pressure conditions, this sampling step might be too large compared to the desired final resolution. In that case, the sampling is set to the value of (final resolution/15).

Line-By-Line (LBL) calculations often require the computation of a large number of line shapes over large to very large spectral intervals. The number of points might then become prohibitive. However, it can be observed that the line profile change slower at a distance from the line centre than it does near the centre. The solution implemented in ASIMUT is the possibility to use a non-uniform grid: near the centre of the line, the optimized step derived with the help of Eq. (3.25) is used; the step is then progressively enlarged as one goes away from the centre. Our algorithm is based on the study of Fomin [37], which decomposes the spectral grid unto a series of sub-intervals.

Let's suppose that the line shape must be calculated on the interval D sufficiently large to cover the line completely. The determination of D will be described in the '*Truncation distance from line center*' sub-section. The limits of the centre zone are defined by $[\nu - C, \nu + C]$ with $C = \frac{2}{3}(\alpha_D + \alpha_L)$, α_D and α_L being the Doppler and Lorentzian widths of the line. The grid is divided into $2L$ portions located at unequal intervals:

$$[\nu - D, \nu - 2^{L-1}C], \dots, [\nu - 2^2C, \nu - 2C], [\nu - 2C, \nu - C], \text{central zone}, \quad (3.26)$$

$$[\nu + C, \nu + 2C], [\nu + 2C, \nu + 2^2C], \dots, [\nu + 2^{L-1}C, \nu + D]$$

The number of intervals is related to D and C through the following expression:

$$2^L \times C = D \quad (3.27)$$

In the central zone, the sampling is set to the optimized value determined with Eq. (3.25). Then in each subsequent interval, the sampling is doubled. This drastically reduces the number of points on which the line profile is calculated, without losing accuracy at the centre of the line.

At the end, the absorption line shape is interpolated to correspond to the sampling used for the determination of the optical depth (OD). The latter is chosen by considering the step asked for the radiance or transmission simulation and an oversampling factor either provided by the user or chosen such that the OD wavenumber step is a factor 10 lower than the final radiance step.

Rejection of lines – Any spectral line whose center intensity is less than a certain absorption level is rejected. This level is by default $A_{ODLimit} = 1.0 \times 10^{-10}$, but can be adjusted by the user with the `[SPn_FENx/ODLimit` parameter. The rejection criteria is based on the assumption that the line is a Voigt line profile and that tests can be done separately on conditions imposed by the Doppler part of the profile and of the Lorentzian part.

For Lorentz profiles, the transmission $T(\nu)$ in one specific layer is defined as:

$$T(\nu) = 1 - A(\nu) = \exp \left[-\frac{S u \gamma_L}{\pi \left[(\nu - \nu_0)^2 + \gamma_L^2 \right]} \right] \quad (3.28)$$

where S is the line intensity (from HITRAN), u is the integrated density in the layer, ν_0 is the position of the line center. This can also be rewritten as :

$$\ln(1 - A(\nu)) = -\frac{S u \gamma_L}{\pi \left[(\nu - \nu_0)^2 + \gamma_L^2 \right]} \quad (3.29)$$

If the line is faint $A \ll 1$ and $\ln(1 - A) \sim -A$, leading to:

$$A(\nu) \approx \frac{S u \gamma_L}{\pi \left[(\nu - \nu_0)^2 + \gamma_L^2 \right]} \quad (3.30)$$

At the center of the line, the intensity is:

$$A_{Lorentz}(\nu_0) \approx \frac{S u}{\pi \gamma_L} \quad (3.31)$$

A Lorentz line having an intensity $A_{Lorentz}(\nu_0)$ lower than the OD limit will not be computed in the corresponding layer.

The same reasoning can be applied on a *Doppler* limited profile, with the respective equations:

$$T(\nu) = 1 - A(\nu) = \exp \left\{ -\frac{Su}{\gamma_D} \sqrt{\frac{\ln 2}{\pi}} \exp \left[-\ln 2 \left(\frac{\nu - \nu_0}{\gamma_D} \right)^2 \right] \right\} \quad (3.32)$$

$$A(\nu) \approx \frac{Su}{\gamma_D} \sqrt{\frac{\ln 2}{\pi}} \exp \left[-\ln 2 \left(\frac{\nu - \nu_0}{\gamma_D} \right)^2 \right] \quad (3.33)$$

$$A_{Doppler}(\nu_0) = \frac{Su}{\gamma_D} \sqrt{\frac{\ln 2}{\pi}} \quad (3.34)$$

The absorption at the center of a Voigt profile is always less than the absorption at the center of both the Doppler and the Lorentz profile from which it is calculated. As a result if the line intensity at the center is less than both limits defined by Equ. (3.31) and (3.34), then the line can be rejected. The criterion of rejection is thus the following:

$$\min(A_{Lorentz}, A_{Doppler}) < A_{ODLimit} \quad (3.35)$$

Truncation distance from the line centre – The determination of the distance from the line centre at which the calculation of the line profile is stopped (D) is based on 2 criteria. Note that the value of D can also be imposed by the user in [MolecLP]FBord option and when χ factor is used, D is defined as the width of the spectral interval on which the χ function is given.

The **first criterion** ensures that the line is not computed for wavenumbers outside a spectral interval defined by a minimum cut-off value imposed on the normalized line profile (before multiplication by the line intensity and the integrated density of the species in the corresponding layer). The reasoning is based on the assumption that the line is a Voigt line profile and that tests can be done separately on conditions imposed by the Doppler part of the profile and of the Lorentzian part.

For a Lorentz profile, intensity will go under the cut-off limit ($A_{cutoff} = 1.0 \times 10^{-6}$) when:

$$(\nu_{cutoff}^{Lorentz} - \nu_0) = \sqrt{\gamma_L \left(\frac{1}{\pi A_{cutoff}} - \gamma_L \right)} \quad (3.36)$$

and for a Doppler profile, the relation becomes:

$$(\nu_{cutoff}^{Doppler} - \nu_0) = \frac{\gamma_D}{\sqrt{\ln 2}} \sqrt{\ln \left(\frac{\sqrt{\frac{\ln 2}{\pi}}}{\gamma_D A_{cutoff}} \right)} \quad (3.37)$$

The interval on which the profile will be calculated is thus limited at:

$$\nu_{cutoff}^{crit1} = \max(\nu_{cutoff}^{Lorentz}, \nu_{cutoff}^{Doppler}) \quad (3.38)$$

The **second criterion** check when the line corrected with the line intensity (from HITRAN, for example) and the integrated density in one layer goes under a second

cut-off limit ($A_{cutoff\text{-intensity}} = 5.0 \times 10^{-8}$). Similar sets of equations are used to define the cut-off position in that case:

$$v_{cutoff}^{crit2} = \max(v_{cutoffintensity}^{Lorentz}, v_{cutoffintensity}^{Doppler}) \quad (3.39)$$

with

$$(v_{cutoffintensity}^{Lorentz} - v_0) = \sqrt{\gamma_L \left(\frac{Su}{\pi A_{cutoffintensity}} - \gamma_L \right)} \quad (3.40)$$

$$(v_{cutoffintensity}^{Doppler} - v_0) = \frac{\gamma_D}{\sqrt{\ln 2}} \sqrt{\ln \left(\frac{\sqrt{\ln 2} Su}{\gamma_D A_{cutoffintensity}} \right)} \quad (3.41)$$

The final definition of the cut-off position is based on the observation that the criteria defined in Equ. (3.38) and (3.39) are upper limits. Finally the cut-off is defined as :

$$v_{cutoff} = \min(v_{cutoff}^{crit2}, v_{cutoff}^{crit2}) \quad (3.42)$$

Wavenumber intervals - The user defines the wavenumber spectral interval on which the simulation/retrieval will be performed. Limits for plotting can also be supplied. This spectral interval will be extended in order to (1) ensure that no information will be lost when performing the convolution [adding W_C = an interval whose width is calculated as being $14 * Res$, where Res is the final resolution of the spectrum]; (2) ensure that all lines near the extremities but not within the chosen spectral interval will be taken into account [adding W_{Select} = at least 25.0 cm^{-1} to both extremities; this value can be extended by using the *[MolecLP]FBord* option for the molecules included in the simulation; When χ factor is used, the width of the spectral interval on which the χ function is given is considered]. These limits are summarized in Figure 4.

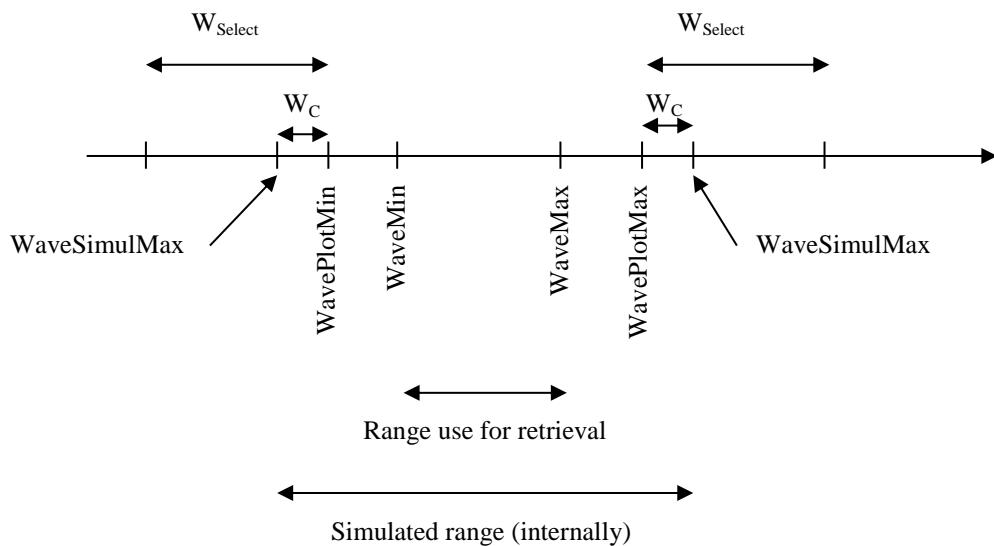


Figure 4: Spectral limits defined and used by ASIMUT

3.2.1.2 Spectral line catalogues

Spectroscopic parameters, such as line position, intensity, and broadening coefficients, as well as absorption cross sections are taken, for practical reasons, from existing spectroscopic databases. ASIMUT can directly import data from the two widely used HITRAN [38] and GEISA [39] databases, but also from the CDSD [54] and CFGL (used by the SFIT community) database whose format is similar to that of HITRAN. However, the program can also read user provided Ascii files in which the line parameters are given. Table 4 describes the parameters which have to be present in the user file. Molecules included in the HITRAN and GEISA databases cover a wide range of atmospheric species, as shown in Table 5 which indicates the molecule ID for each possible molecule. The isotopologue ID has been chosen as the one defined in HITRAN, directly in relation to their relative abundance in Earth atmosphere. For some species, the GEISA database reports more isotopologues than HITRAN. This is the case for the $^{18}\text{O}^{13}\text{C}^{18}\text{O}$, $^{16}\text{O}^{13}\text{C}^{34}\text{S}$, $^{17}\text{O}^{12}\text{C}^{32}\text{S}$, and $^{12}\text{C}^{13}\text{CH}_6$ species, which are included in ASIMUT. The older versions of the HITRAN databases with a slightly different format than the current version are also supported.

Parameter	Symbol	Unit
Molecule ID	mol	-
Isotopologue ID	iso	-
Central wavenumber	ν_{ij}	cm^{-1}
Line intensity	$S_{ij}(T_0)$	$\text{cm}^{-1}/(\text{molec} \cdot \text{cm}^2)$
Self-broadening pressure half width	γ_L^{self}	$\text{cm}^{-1} \text{atm}^{-1}$
Air broadening pressure half width	$\gamma_L^{foreign}$	$\text{cm}^{-1} \text{atm}^{-1}$
Temperature dependence of the self-broadening coefficient	n_{self}	-
Temperature dependence of the air broadening coefficient	n_{air}	-
Self-pressure wavenumber shift	δ_{self}	$\text{cm}^{-1} \text{atm}^{-1}$
Air pressure wavenumber shift	$\delta_{foreign}$	$\text{cm}^{-1} \text{atm}^{-1}$
Temperature dependence of the self-pressure shift	n_{self}^s	-
Temperature dependence of the air pressure shift	$n_{foreign}^s$	-
Collision narrowing parameter	β°	$\text{cm}^{-1} \text{atm}^{-1}$
Energy of the lower transition level	E_j	cm^{-1}

Table 4: Line parameters required by ASIMUT

3.2.1.3 Line Mixing

[to be completed]

For the moment : only for CO2 with the 1st order approximation method both under WINDOWS and UNIX; only full mixing method under WINDOWS.

The full mixing requires the inclusion of the LAPACK (in C) library.

Ref : Lamouroux et al.

3.2.1.4 [Molecules] options

[Molecules]fileHITRAN : Localization of the general HITRAN file (used if no file is specified for the molecule under [MolecLP]File and [MolecLP]Type is 'hitran'. See Annex 2 for the description of the HITRAN format.

[Molecules]fileGEISA : Localization of the general **GEISA**file (used if no file is specified for the molecule under [MolecLP]File and [MolecLP]Type is 'geisa'. See Annex 3 for the description of the HITRAN format.

[Molecules]PartitionFct : Method used to calculate the Partition function: TIPS matrix or TPOLY temperature polynomials (equ. 8).

3.2.1.5 [xxxLP] options

In the name of these sections, xxx represent the name of the molecule (ex: 'O3', 'CO2', 'NewO2', 'formaldehyde', ...). 'LP' stands for Line Parameter.

[xxxLP]DBname : It is possible to use a name xxx for the molecule internally defined in ASIMUT, but non-existing in the database usually used. For example, one can used the 'O3isol' name : it will be recognized by ASIMUT, but not in the HITRAN database which only recognizes 'O3'. **[xxxLP]DBname** is the name to be found in the database.

[xxxLP]ATMname : It is possible to use a name xxx for the molecule internally defined in ASIMUT, but non-existing in the atmospheric files usually used. For example, one can used the 'O3isol' name : it will be recognized by ASIMUT, but not in the US Standard database which only recognizes 'O3'. **[xxxLP]ATMname** is the name to be found in the atmospheric file.

[xxxLP]File : name of the file containing the spectroscopic parameters

[xxxLP]Type : type of the file containing the spectroscopic parameters: 'hitran', 'geisa', 'cfgl' for the main databases. See Annex 2 and Annex 3 for the description of the HITRAN and GEISA formats respectively. It can also be 'ascii' or 'grat'. The latter is the internal format of ASIMUT (binary code).

If 'ascii' is selected, the following columns must be given: see Table 4.

[xxxLP]isotope : definition of the isotopologue of the molecule. If '999', all isotopologues will be considered. If '999-xxx', take all isotopologues except xxx.

Be sure that the spectroscopic parameters are then corrected by the isotopic ratios (this is the case for 'hitran', 'geisa', 'cfgl'. For 'grat' files it is also the case if the files were created with the '999' option).

To specify a given isotopologue, use the HITRAN denomination (see **Error! Reference source not found.**).

Examples: (1) all isotopologues of O3 together

```
[03LP]
Isotope = 999
```

...

(2) two isotopologues of O3 (1st and 3d ones)
[03i1LP]
Isotope = 666

```

...
[03i3LP]
Isotope = 686
...
(3) all isotopologues of O3 except the main one
[03i1LP]
Isotope = 999-666
...

```

[xxxLP]isoRatio : Isotopic ratio for the species ($0 < \dots < 1.0$). If no number is given, then it takes the isotopic ratio on Earth (HITRAN values). Not applicable for isotope='999' (always = 1.0).

[xxxLP]limite : Limit for the intensity value. All line with intensity < limit, will be rejected

[xxxLP]model : Model for a priori profile (can be different than the general value). See 3.1.2.1[AtmosphericModels] options) for a complete description of the possible values.

[xxxLP]ATMfile : File containing the atmospheric vertical profile for the species. Must be given if 'model' for the species is set to 0, and species not present in the [AtmosphericModels]model file (or if this profile should supersede the one in that file).

Format of file: 2 col

```

Col 1 : z (km)
Col 2 : vmr (ppm)
```

The following parameters defines the Line profile to apply for this molecule:

[xxxLP]LPprofile: Type of line profile applied. Different profiles can be selected : VoigtPade, VoigtHumlicek, VoigtFaddeeva, Rautian, Galatry, Lorentz

[xxxLP]FBord: Spectral interval on which the line profile is calculated (cm^{-1}). Must be $< 400 \text{ cm}^{-1}$. If not specified, internal value (25 cm^{-1}) of ASIMUT is used.

[xxxLP]ChiFactor : Name of file containing a Chi Factor: 2 col

```

Col 1 = delta wavenumber ( $\text{cm}^{-1}$ ) = nu-nucenter
Col 2 = Chi fct (max=1 at center)
```

The Chi factor must be given for negative and positive delta wavenumber (can be asymmetric)

[xxxLP]Collision : Collision parameter used for Rautian profile

[xxxLP]shift : Constant shift applied on all lines (cm^{-1})

The following parameters supersede the values found in the spectroscopic file:

[xxxLP]gammaself: Self broadening parameter ($\text{cm}^{-1}/\text{atm}$)

- 2 Do nothing (take HITRAN/ GEISA values as they are)
- 1 Take value of GammaAir
- 0.0 All GammaSelf=0.0
- >0.0 Default value, when HITRAN/ GEISA are 0.0

[xxxLP]GammaAir: Default value, when HITRAN/ GEISA are 0.0 ($\text{cm}^{-1}/\text{atm}$)

[xxxLP]ShiftSelf: Default value, when HITRAN/ GEISA are 0.0 ($\text{cm}^{-1}/\text{atm}$)

[xxxLP]ShiftAir: Default value, when HITRAN/ GEISA are 0.0 ($\text{cm}^{-1}/\text{atm}$)

[xxxLP]ShiftSelf: Default value, when HITRAN/ GEISA are 0.0 ($\text{cm}^{-1}/\text{atm}$)

[xxxLP]nSelf: Default value, when HITRAN/ GEISA are 0.0

[xxxLP]nAir: Default value, when HITRAN/ GEISA are 0.0

[xxxLP]zfact, fact: factors that can be applied on the vertical profile read from the atmosphere file.

[xxxLP]LineMixing: Indicates if Line Mixing has to be included and what type of line mixing.

0/n/no No line mixing

1/y/yes Line mixing with 1st order approximation

full Line mixing: full treatment [only possible if the FULLLINEMIXING option is used at compilation, and if the LAPACK library has been added to the project] [For the moment: only implemented for WINDOWS runs]

[xxxLP]LineMixingDir: Path where the data used for the line mixing calculation are located (relative to *[Directories]dirLP*)
(final path will be *[Directories]dirLP\LineMixingPath*).

3.2.2 Cross sections

Heavy molecules give rise to generally dense absorption spectra, which can not be resolved into single transitions. Similarly, absorption in the visible and the ultraviolet is also characterized by broad unresolved structures. In those cases, one defines the absorption through the cross section, which depends on temperature and pressure. ASIMUT can import cross section data from the HITRAN[38] and GEISA[39] databases, as well as data given by the user. As those data may be rather scarce in some cases in terms of different temperature and pressure conditions, it was decided to use the data corresponding to the nearest couple (T, P) present in the databases without any interpolation. The contribution of the absorption cross sections to the absorption coefficient of a layer characterized by temperature T and total pressure P is the following:

$$\alpha_{cs}(\nu, P, T) = \sum_j n_j \sigma_j(\nu, P, T, p_j) \quad (3.43)$$

where σ_j is the absorption cross section of species j , whose cumulated density in the layer is n_j (molec/cm²).

3.2.2.1 Cross sections catalogues

Absorption cross sections can be read from the HITRAN and GEISA compilations, as well as from user defined files, in which different data columns can be associated to different temperature and pressure conditions. The import of cross section data is not limited to the species found in the HITRAN and GEISA databases.

Species	Molecule ID
H ₂ O, CO ₂ , O ₃ , N ₂ O, CO, CH ₄ , O ₂ , NO, SO ₂ , NO ₂ , NH ₃ , HNO ₃ , OH, HF, HCl, HBr, HI, ClO, OCS, H ₂ CO, HOCl, N ₂ , HCN, CH ₃ Cl, H ₂ O ₂ , C ₂ H ₂ , C ₂ H ₆ , PH ₃ , COF ₂ , SF ₆ , H ₂ S, HCOOH, HO ₂ , O, ClONO ₂ , NO+, HOBr, C ₂ H ₄ , CH ₃ OH, CH ₃ Br, CH ₃ CN, CF ₄	As in HITRAN (from 1 to 42)
GeH ₄ , C ₃ H ₈ , C ₂ N ₂ , C ₄ H ₂ , HC ₃ N, C ₃ H ₄	From 43 to 48 (molecules in GEISA, but not in HITRAN)
HONO, HO ₂ NO ₂ , N ₂ O ₅ , CH ₃ F, CCL ₂ F ₂ , CCL ₃ F, CH ₃ CCL ₃ , CCL ₄ , COCLF, CHF ₂ CL, COCL ₂ , CH ₃ I, CHCL ₂ F, OCIO, F134A, F142B, CFC113, F141B, C ₂ H ₆ PL, PAN	From 49 to 68 (molecules in CFGL, but not in HITRAN)

Table 5: Molecules recognized by ASIMUT

3.2.2.2 [xxxXS] options

In the name of these sections, xxx represent the name of the molecule (ex: 'O3', 'CO2', 'NewO2', 'formaldehyde', ...). 'XS' stands for Cross Section.

[xxxXS]ATMname : name to be found in the atmospheric file. Must be given.

[xxxXS]File : name of the file containing the spectroscopic parameters. Must be given.

[xxxXS]Type : type of the file containing the cross sections data: 'hitran', 'geisa', 'ascii'.

If 'ascii' is selected, the following columns must be given: wavenumber [cm⁻¹], N columns [cm²/molec], with N the number of different laboratory conditions, which will be described by the next parameters (*Temperature*, *Pressure*, *air*, *resolution*, *resolUnit*).

The following parameters describe the laboratory conditions under which the data in the different columns of the **[xxxXS]File** have been obtained:

[xxxXS]Temperature : Temperature (K) of each column of the **[xxxXS]File**.

[xxxXS]pressure : pressure (torr) of each column of the **[xxxXS]File**.

[xxxXS]air : indicates if the wavenumber are expressed in air (1) or vacuum (0).

[xxxXS]resolution : resolution of each column of the **[xxxXS]File**.

[xxxXS]resolUnit : unit of the resolution : cm⁻¹ (1) or nm (0).

[xxxLP]zfact, fact: factors that can be applied on the vertical profile read from the atmosphere file.

3.2.3 Continua

For some gases such as N₂, O₂, CO₂, and H₂O, broad absorption features cover wide spectral regions in addition to single absorption lines. These continua of absorption are due to different physical phenomenon: Collision-induced absorption bands resulting from the interaction of pairs of molecules such as O₂-O₂, N₂-O₂, N₂-N₂, superposition of line wings in the case of H₂O and CO₂. Different molecular continua are included in ASIMUT. They will be described in details hereinafter. In general, they follow the MT_CKD_1.2 formalism [20] with modifications introduced to take into account new laboratory measurements or the possibility to simulate atmospheres other than Earth.

3.2.3.1 O₂ continua

Fundamental band at 6.4 μm

Thibault et al. [40] performed new measurements of the O₂-O₂ and O₂-N₂ absorption coefficients in the 0-20 atm and 193-293 K pressure and temperature ranges. Moreover, they made use of available experimental data in the 190-360 K temperature range to build a simple empirical model. From measurements of the O₂-O₂ and O₂-N₂ collision-induced absorption, they calculated the O₂-air absorption coefficient:

$$B_{O_2\text{-}air}(\nu, T) = \frac{P_{O_2}}{P_{tot}} B_{O_2\text{-}O_2}(\nu, T) + \frac{P_{N_2}}{P_{tot}} B_{O_2\text{-}N_2}(\nu, T) \quad (3.44)$$

The temperature and wavenumber dependences were then modeled using the simple empirical law

$$B_{O_2\text{-}air}(\nu, T) = B_{O_2\text{-}air}^0(\nu) \exp\left[\beta_{O_2\text{-}air}(\nu)\left(\frac{1}{T_R} - \frac{1}{T}\right)\right] \quad (3.45)$$

in which the reference temperature $T_R = 296$ K was retained. The values given in [40] are expressed in Amagat, $B_{O_2\text{-}air}(\nu, T)$ expression has therefore to be slightly modified to be compatible with the units used in ASIMUT. Finally, the absorption coefficient for the layer at altitude z is given by:

$$\alpha_{O_2\text{-}air}(\nu, z) = n_{O_2} \frac{P}{P_0} \frac{T_0}{T} \frac{1}{N_L} B_{O_2\text{-}air}(\nu, T(z)) \quad (3.46)$$

with P_0 and T_0 the standard pressure and temperature (1013.25 hPa, 273.15 K), N_L the Loschmidt number ($2.686\ 777 \times 10^{19} \text{ cm}^{-3}$), and n_{O_2} is the O₂ cumulated density in the layer z .

Near-infrared and visible bands

Two different calculations of the collision-induced absorption coefficients for the near-infrared and visible O₂ bands are implemented in ASIMUT. The first one is based on the MT_CKD_1.2 continua implemented in the latest version of LBLRTM, and the second is based on more recent measurements of these bands [41].

The 1.27 μm band is the most intense of the three near-infrared collision-induced O₂ bands. The MT_CKD_1.2 is based on measurements performed by Maté et al. [42] using sample densities from 1 to 10 times that of ideal gas under standard conditions at temperatures of 253, 273, and 296 K for pure O₂ and O₂/N₂ mixtures. From their measurements, they derived the collision-induced absorption coefficient for a mixture

of 79% N₂ and 21% O₂. These data are however corrected to take into account observations of the 1.27 μm band in atmospheric spectra [43].

In the same study, the 1.06 μm continuum was fitted using the sum of two Lorentzians centered at 9375 and 9439 cm⁻¹, with exponential damping on their low-wavenumber sides. The MT_CKD_1.2 makes also use of the data obtained by Greenblatt under high pressure conditions [44].

The second implementation is based on new measurements [41], who reported values for all O₂-O₂ collision-induced absorption structures existing between 7000 and 30000 cm⁻¹.

Herzberg and Wulf continua

Measurements of Bernath et al. [45] and of Fally et al. [46] of the Herzberg continuum and of the Wulf bands of the oxygen molecule have been implemented. They cover the wide spectral range from 34000 cm⁻¹ to 41500 cm⁻¹. These measurements were performed at a resolution of 2 cm⁻¹. To obtain the absorption cross section (cm² molecule⁻¹) of the O₂ continuum in this region, the following expression has been used:

$$\sigma_{O_2-O_2}(\nu) = \sigma_{Herzberg}(\nu) + [Wulf_{O_2} + g Wulf_{N_2}] n_{O_2} \quad (3.47)$$

with

$$g = \frac{n_{N_2}}{n_{O_2}} \quad (3.48)$$

where n_{N_2} and n_{O_2} are the densities of respectively N₂ and O₂. This expression is valid up to 41500 cm⁻¹. However, as stated in [46], the uncertainty on the Herzberg and Wulf continua is high above 41000 cm⁻¹. The validity of Eq. (3.47) has been verified up to 41227 cm⁻¹. However, the Herzberg continuum extends well above 42000 cm⁻¹. The formalism used in LBLRTM, based on Yoshino's data [47] has been considered above 41227 cm⁻¹.

3.2.3.2 N₂ continua

Collision-induced band at 350 cm⁻¹

Based on the works of Borysov and Frommhold [48] and of Boissoles et al. [49], the modeling of the collision-induced absorption coefficient in air takes into account the temperature dependence of the absorption.

Collision-induced absorption in the fundamental band

Lafferty et al. [50] carried out measurements of collision-induced absorption of pure nitrogen in the fundamental band near 4.3 μm. The measurements were performed under different pressure (0-10 atm) and temperature (230-300 K) conditions. Using available literature data in the 190-300 K temperature range, Lafferty et al. built a simple empirical model to compute the N₂-N₂ absorption. The dependence of the absorption for pure N₂ towards temperature can be represented by the following expression:

$$B_{N_2-N_2}(\nu, T) = B_{N_2-N_2}^0(\nu) \cdot \exp \left[\beta_{N_2-N_2}(\nu) \left(\frac{1}{T_0} - \frac{1}{T} \right) \right] \quad (3.49)$$

where $B_{N_2-N_2}^0(\nu)$ is the absorption at the reference temperature T_0 that has been chosen as 296 K, $\beta_{N_2-N_2}(\nu)$ is the temperature coefficient of the collision-induced absorption.

The relative efficiency $E_{O_2/N_2}^{N_2}$ of the colliding partners O₂ and N₂, defined by [51]:

$$E_{O_2/N_2}^{N_2}(\nu, T) = \frac{B_{N_2-O_2}(\nu, T)}{B_{N_2-N_2}(\nu, T)} \quad (3.50)$$

is practically independent of wavenumber. The N₂-air collision-induced absorption can be written as:

$$B_{N_2-air}(\nu, T) = \left[\frac{p_{N_2}}{p_{tot}} + \frac{p_{O_2}}{p_{tot}} E_{O_2/N_2}^{N_2}(T) \right] B_{N_2-N_2}(\nu, T) \quad (3.51)$$

where p_{N_2} and p_{O_2} are the partial pressures of N₂ and O₂ respectively. Menoux et al. [51] showed that the relative efficiencies of collisions with O₂ and N₂ can be accurately modeled by the following wavenumber independent relation:

$$E_{O_2/N_2}^{N_2}(T) = 1.294 - 0.4545 \frac{T}{T_0} \quad (3.52)$$

Finally, the N₂-air collision-induced absorption is obtained by inserting Eq. (3.52) into Eq. (3.51) and using the Lafferty et al. [50] model for the $B_{N_2-N_2}(\nu, T)$ absorption.

3.2.3.3 CO₂ and H₂O continua

The formalism of MT_CKD_1.2 has been implemented in ASIMUT. The modeling of these continua has been extensively described elsewhere [52].

3.2.3.4 [Continua] options

[Continua]H2O : Formalism for H₂O continuum

mtckd12	MT_CKD_1.2
mtckd13	MT_CKD_1.3 (same as 1.2)
mtckd25	MT_CKD_2.5

[Continua]CO2 : Formalism for CO₂ continuum

mtckd12	MT_CKD_1.2
mtckd13	MT_CKD_1.3 (same as 1.2)
mtckd25	MT_CKD_2.5

[Continua]O2 : Formalism for O₂ continuum

mtckd12	MT_CKD_1.2
mtckd13	MT_CKD_1.3 (same as 1.2)
mtckd25	MT_CKD_2.5
br	BR (IASB/ULB/REIMS) measurements

[Continua]N2 : Formalism for N₂ continuum

mtckd12	MT_CKD_1.2
mtckd13	MT_CKD_1.3 (same as 1.2)
mtckd13Cor	MT_CKD_1.3 + corrections
mtckd25	MT_CKD_2.5

3.2.4 Rayleigh scattering

Molecular scattering is certainly the strongest mechanism causing atmospheric extinction of solar radiation in cloudless atmospheres. The Rayleigh optical depth $\alpha_R(\nu, z)$ at wavenumber ν , for the layer at the altitude z , is given by:

$$\alpha_R(\nu, z) = N(z) \sigma(\nu, z) \quad (3.53)$$

where $N(z)$ is the molecular number density of air in the layer, and $\sigma(\nu, z)$ is the total Rayleigh scattering cross section per molecule, which may be expressed as:

$$\sigma(\nu, z) = \frac{24\pi^3}{\lambda^4 N(z)^2} \frac{[n(\nu, z)^2 - 1]^2}{[n(\nu, z)^2 + 2]^2} F(\nu, z) \quad (3.54)$$

where $n(\nu, z)$ is the refractive index of air at ν and z ; $F(\nu, z)$ is the King factor for the depolarization of air at ν and z , that accounts for the anisotropic properties of air molecules: anisotropic molecules scatter more radiation at 90° than isotropic ones with the same index of refraction. The F factor is calculated as the weighted sum of the partial contributions due to the molecules N_2 , O_2 , Ar , and CO_2 , considering that each contribution varies as a function of wavelength. The weights are given by the volume mixing ratios of the respective atmospheric constituents [24].

In parallel to this implementation of the Rayleigh extinction (*Ciddor*), the user may select other methods:

LBLRTM: the calculation of the Rayleigh cross section is very basic and parameterized;

SHDOM : extract from the SHDOM software package (similar to LIDORT, but developed for Mars). It only contains the contribution from CO_2 and O_2 which are the main constituents of the Martian atmosphere;

SNEEP : based on the formulas of Sneep and Ubachs [53] giving Rayleigh and King contributions for different species (CO_2 , O_2 , N_2 , ...);

Two other very basic parameterizations (VENUS and MARS).

3.2.4.1 [Continua] options

[Continua]Rayleigh : Formalism for Rayleigh scattering

mtckd	MT_CKD_1.2
Hansen	Hansen and Travis, 1974 (from pressure)
Shdom	shdom (valid only for Mars/Venus)
Sneep	Sneep and Ubachs (JQSRT 92, 293-310, 2005)
Ciddor	Ciddor (AO, 35, 1566-1573, 1996)

[Continua] RayleighDepolarisation: Rayleigh depolarisation ratio.

Default value for Earth: Young, AO, 1980

Default value for Mars & Venus: 0.0774 (SHDOM model)

If =0.0: no scattering effect (but still absorption)

3.2.5 Aerosols

Aerosols are included in the ASIMUT code, either as extinction (ASIMUT) or full absorbing and scattering species (ALVL, through the call to (V)LIDORT).

Different types of parameterization are included. The aerosols calculated by (V)LIDORT are defined just as another type of aerosols. In that case ASIMUT is just reading the input parameters and letting (V)LIDORT do the calculations (or call the SPHER/T-MATRIX). However, as they are ‘aerosols’, they have access to the whole range of routines already defined in ASIMUT dealing with aerosols, such as:

- 1) multiplying the profile by a factor (zFact and Fact in the .INP file) or imposing a total optical depth (TotalOpacity);
- 2) saving in the .out files and in the optical depths files;
- 3) fitting of the ‘aerosol densities’, as a profile or as a column + the full Rodgers treatment (including full Sa matrix).

In one single run, different aerosols of different types can be defined, the classical ones will be calculated by ASIMUT and the ‘Lidort’ or scattering ones by (V)LIDORT. All can be fitted (for the moment only their ‘density’ or OD).

The transmittance corresponding to the aerosol extinction in layer L is calculated by

$$T_{aer,L} = \exp(-\tau_{aer,L}) \quad (3.55)$$

where $\tau_{aer,L}$ is the aerosol optical thickness (=optical depth).

Angström formalism (Type= Angstrom)

The aerosol optical thickness in the layer L is calculated using:

$$\tau_{aer,L} = \tau_{\lambda0,L} \left(\frac{\lambda_0}{\lambda} \right)^\alpha = \tau_{\lambda0,L} \left(\frac{\nu}{\nu_0} \right)^\alpha = \kappa_{\lambda0,L} \Delta s_L \left(\frac{\nu}{\nu_0} \right)^\alpha \quad (3.56)$$

with ν_0 , the reference wavenumber,

$\tau_{\lambda0,L}$ the optical depth at this wavenumber and in layer L of width Δs_L (km),

$\kappa_{\lambda0,L}$ is the extinction coefficient of layer L (km^{-1}) given in *kext* input parameters, defined on scale *zFact*.

α the Angström Coefficient

Modified Angström formalism (Type= ModAngstrom)

The aerosol optical thickness in the layer L is calculated using:

$$\tau_{aer,L} = \tau_{\lambda0,L} \cdot f(\nu) = \kappa_{\lambda0,L} \Delta s_L \cdot f(\nu) \quad (3.57)$$

with ν_0 , the reference wavenumber,

$\tau_{\lambda0,L}$ the optical depth at this wavenumber and in layer L of width Δs_L (km),

$\kappa_{\lambda0,L}$ is the extinction coefficient of layer L (km^{-1}) and given in *kext* input parameters, defined on scale *zFact*,

$f(\nu)$ is a function representing the dependence of the optical depth with wavelength. Values are given in the file *QextFileName* (2 columns: wavenumber-cm-1, function values). These values will be interpolated on the

spectrum wavenumber scale, with the condition that at ν_0 it equals 1.0: $f(\nu_0) = 1.0$

Conrath formalism (Type= Conrath)

The aerosol optical thickness in the layer L is calculated using:

$$\tau_{aer,L} = q_0 \exp \left[\gamma \left(1 - \frac{p_{surface}}{p_L} \right) \right] n_L \Delta s_L \left(\frac{\lambda_0}{\lambda} \right)^\alpha \quad (3.58)$$

with γ , a parameter describing the aerosols (for Mars: $\gamma=0.007$ during a dust storm, $\gamma=0.05$ during polar night, $\gamma=0.03$ in general), $p_{surface}$ and p_L , respectively, the pressure at the surface and in the layer L, n_L the density in the layer L, and Δs_L the path length within layer. The value of q_0 is calculated from the ‘TotalOpacity’ parameter.

It can also be expressed in terms of scale height:

$$\tau_{aer,L} = q_0 \exp \left[\gamma \left(1 - \exp \left(\frac{z_L}{sch_L} \right) \right) \right] n_L \Delta s_L \left(\frac{\lambda_0}{\lambda} \right)^\alpha \quad (3.59)$$

with : $sch_L = \frac{R_{gas} T_L}{\mu g_L}$, where T_L is the temperature in the layer L, m is the molecular

weight of dry air, and $g_L = \frac{g_{Surface}}{\left(1 + \frac{z_L}{R_p} \right)^2}$ is the acceleration of gravity at z_L (R_p is the radius of the planet, g_0 the acceleration of gravity at the surface).

Qext formalism (Type= Qext)

The aerosol optical thickness in the layer L is calculated using:

$$\tau_{aer,L} = N_{aer,L} Q_{ext} \pi r^2 \Delta s_L \quad (3.60)$$

where $N_{aer,L}$ is the particule density (part/m^3) of the aerosol in layer L, r_{eff} is the effective radius of the aerosols (mm), Q_{ext} is the extinction calculated by Rayleigh or Mie theory (must be provided by the user).

Lidort formalism

If the SSA, OD, and the aerosol scattering matrix coefficients are not given, then ASIMUT will read the supplied Aerosol file (*fileAerosols*) and will call SPHER/TMATRIX to calculate those quantities.

The aerosol optical thickness in the layer L is calculated using the cross section (XS) which is read from the file *fileAerosolOD*, using:

$$\tau_{aer,L} = XS_{fromFile} N_{aer,L} \Delta s_L \quad (3.61)$$

with $N_{aer,L}$ is the particule density (part/m^3) of the aerosol in layer L.

If *kext* are specified, then values of the aerosol optical depth for layer L are normalized to the $\kappa_{\lambda0,L}$ value, which is the extinction coefficient of layer L (km^{-1}) and given in *kext* input parameters, defined on scale *zFact*.

LidortG formalism (Henyey Greenstein)

One file containing SSA, g , and OD wrt wavenumber is read. In that case, the Legendre coefficients are calculated from the g parameters using the Henyey Greenstein formula.

The aerosol optical thickness in the layer L is calculated using the OD which is read from the file *fileAerosolOD*, using:

$$\tau_{aer,L} = OD_{fromFile} N_{aer,L} \Delta s_L \quad (3.62)$$

with $N_{aer,L}$ is the particule density (part/m³) of the aerosol in layer L.

If *kext* are specified, then values of the aerosol optical depth for layer L are normalized to the $\kappa_{\lambda0,L}$ value, which is the extinction coefficient of layer L (km⁻¹) and given in *kext* input parameters, defined on scale *zFact*.

3.2.5.1 [xxxAER] options

In the name of these sections, xxx represent the name of the aerosol (ex: 'Model', 'soot', ...). 'AER' stands for AERosol.

[xxxAER] Type : Type of aerosol:

Angstrom	Angstrom formalism
ModAngstrom	Modified Angstrom formalism
Qext	From given Qext , Dirac distribution
Conrath	Conrath formalism
Lidort	(V) LIDORT formalism
LidortG	(V) LIDORT formalism

[xxxAER] TotalOpacity: Total opacity of aerosols integrated on the whole atmosphere (Lidort, Conrath & Angstrom formalism)

If *TotalOpacity* ≤ 0 : not taken into account

Must be given for Conrath formalism

[xxxLP] zfact, fact: factors that can be applied on the vertical profile read from the atmosphere file.

[xxxAER] kextType: For Angstrom formalism: Type of extinction coefficients (km⁻¹)

Values [v1...] are given in *kext*
File Values are given in a file (filename=*kext*)

[xxxAER] kext: For Angstrom formalism:

[v1...] Extinction coefficients (km⁻¹) corresponding to the altitudes given in *zfact*
Filename Filename containing 2 col= z, *kext* [in *dirAerosol*]

[xxxAER] angstromCoeff: Value of the Angstrom coefficient

[xxxAER] angstromNuRef: Value of the reference wavenumber used in the Angstrom formalism (cm⁻¹). If not given, then it is taken as the mid value of wavenumber of the FEN interval

[xxxAER] Gamma: Gamma parameter in the Conrath formalism

[xxxAER] reff: Effective radius (μm)

[xxxxAER] Naer: For Qext and Lidort(G) formalisms: Densities of aerosols (part/cm³) (corresponding to the altitudes given in zFact, see XXX). This parameter is mandatory for Qext formalism.

[xxxxAER] QextFileName: File containing the Qext values
2 col : nu(cm-1) Qext
Qext must be calculated for the given reff
[in dirAerosol]

3.3 Radiative transfer model

3.3.1 Determination of the Radiance

The radiative transfer calculation in ASIMUT is performed in each layer following the radiation path. The radiation exiting one layer $I_{k+1}(\nu)$ is obtained from the radiation entering it $I_k(\nu)$, through

$$I_{k+1}(\nu) = I_k(\nu)t_k(\nu) + B_{eff}(\nu, k)[1 - t_k(\nu)] \quad (3.63)$$

where $B_{eff}(\nu, k)$ is the emission function for the layer k , and t_k is the transmittance of the layer:

$$t_k(\nu) = e^{-\tau} = e^{-\sum_i \alpha_i(\nu) n_i} \quad (3.64)$$

where the sum is taken over all absorbing processes occurring in the layer; α_i are the absorption coefficients ($\text{cm}^2 \text{ molecule}^{-1}$) and n_i are the cumulated densities in the layer (molecule cm^{-2}).

Different contributions to the Radiance are considered:

- Direct contribution from the source;
- Thermal emission from the surface;
- Thermal emission from the atmosphere;
- Reflection of this thermal atmospheric emission on the surface;
- Reflection of the Sun on the surface (Lambertian).

3.3.2 Atmosphere emission

The emission from each layer is given by:

$$I_{emis}(\text{layer } l) = (1 - t_l)B_{eff}(\bar{T}_l, T_{l+1}, \tau_l) \quad (3.65)$$

where τ_l is the optical depth of layer l , t_l is the transmittance of layer l (Equ. 3.64) and T_{l+1} is temperature of the upper boundary of the layer (wrt to the light propagation direction) and \bar{T}_l is the ‘average’ temperature of layer l .

a) **Old version (‘Bm’)** : emission is calculated as a Planck function at the ‘average’ temperature of the layer – this is in fact only valid if the optical depth of the layer is small enough

$$B_{eff}(\bar{T}_l, T_{l+1}, \tau_l) = B(\bar{T}_l) \quad (3.66)$$

b) **Tau dependent :**

[ref : Clough et al., IEEE Trans. Geosc. and Remote sensing, 44 (5), 1308-1323 (2006)]

$$B_{eff}(\bar{T}_l, T_{l+1}, \tau_l) = B(\bar{T}_l) + [B(T_{l+1}) - B(\bar{T}_l)]F(\tau_l) \quad (3.67)$$

$$F(\tau_l) = 1 - 2 \left[\frac{1}{\tau_l} - \frac{t_l}{1 - t_l} \right] \quad (3.68)$$

For small optical depth, $F(\tau_l) \rightarrow \frac{\tau_l}{6}$

c) Padé 2 parameters

[ref : Clough et al, JGR, 97(D14),15761-15785 (1992)]

$$B_{eff}(\bar{T}_l, T_{l+1}, \tau_l) = \left[B(\bar{T}_l) + (a\tau_l + b\tau_l^2) B(T_{l+1}) \right] / (1 + a\tau_l + b\tau_l^2) \quad (3.69)$$

with $a=0.193$ and $b=0.013$.

d) Padé 1 parameter

[ref : Clough et al, JGR, 97(D14),15761-15785 (1992)]

$$B_{eff}(\bar{T}_l, T_{l+1}, \tau_l) = \left[B(\bar{T}_l) + (a\tau_l) B(T_{l+1}) \right] / (1 + a\tau_l) \quad (3.70)$$

with $a=0.2$.

3.3.3 Surface contribution

The surface is characterized by a temperature (T_s) and an emissivity. It is considered Lambertian (at least when using only ASIMUT), different types of surface (Hapke, ...) can be selected when using Lidort.

When the reflection of the solar radiation on the surface is included, the lower boundary condition (at the surface) is the following:

$$I^{BC}(z_{BC}) = \frac{a}{\pi} E^\odot \mu_0 \tau(z_t, z_b, \mu_0) + \varepsilon B(T_{Surface}) \quad (3.71)$$

with E^\odot = solar irradiance at the top of the atmosphere of the planet
 $= \pi \cdot (r/R_{P-S})^2 \cdot I_{Sun}$, where $r R_{P-S}$ = distance Planet-Sun, r = radius of the Sun (at the photosphere) (6.96×10^8 m) and I_{Sun} is the Sun radiance (see Annex 4);

$$\mu_0 = \cos(sza);$$

$\tau(z_t, z_b, \mu_0)$ is the atmosphere transmittance from the top of the atmosphere (z_t)

to the lower boundary (z_b) along the Sun Line of path defined μ_0 ;

ε = emissivity of the planet;

a = albedo of the planet (if not specified : $a = 1 - \varepsilon$)

The surface contribution for the atmospheric emission is characterized either as a Lambertian or specular surface reflection. In the Lambertian case, the surface BRDF is independent of the angle and can be expressed simply in terms of albedo which, when multiplied by the downwelling radiance computed along the diffusivity ray (secant = 1.66), provides a good approximation for the upwelling isotropic radiance

at the surface. The optical depth of one layer along the diffusivity ray is obtained by multiplying the nadir (off-nadir) optical depth by the secant of the diffusivity angle.

In ASIMUT the user can specify the type of surface reflectivity used for the reflection of the radiance emitted by the atmosphere:

LambertSurfAtmEmi =yes / no

3.3.4 Determination of the Transmittances

The transmittance associated to the calculated Radiance is determined using :

$$\text{Transmittance} = \frac{\text{Radiance}}{I_0} \quad (3.72)$$

in which I_0 is the radiance from the source. When looking at the Sun (solar occultation), I_0 is the radiance of the Sun. When looking down at the planet' surface (nadir and off-nadir modes), I_0 is the radiance of the surface, if no solar reflection is taken into account.

If solar reflection on the surface is included, then the radiance is calculated using:

$$\text{Transmittance} = \frac{\text{Radiance} \times \pi}{E\Theta} = \text{Radiance Factor} \quad (3.73)$$

3.3.5 Brightness temperature

The radiance can also be saved as Brightness Temperature, which is defined as the temperature, in Kelvin, of a blackbody that emits the observed radiance. It can be expressed as:

$$T_B(K) = \frac{c_2 \nu}{\ln \left[1 + \frac{(c_1 \nu^3)}{\text{Radiance}(\nu)} \right]} \quad (3.74)$$

where $c_1 = 1.191\ 042\ 7 \times 10^{-8}$ (W m⁻² sr⁻¹ cm⁻⁴) is the first radiation constant for spectral radiance, and $c_2 = 1.438\ 775\ 2$ (cm K⁻¹) is the second radiation constant for spectral radiance.

3.3.5.1 [Solar] options

[Solar]FileSolar : Filename of solar irradiance.

Format for FileSolar :

For ex: irradiance calculated by ACE_irradiance.m

col 1: wavenumber (cm⁻¹)

col 2: IRRADIANCE at top of Earth atmosphere (W/cm²/ (cm⁻¹)).

Irradiance(v) = $\pi \cdot (r/R)^2 \cdot B(v, T_s)$

With T_s : Temperature of Sun photosphere

r : Sun radius

R : distance between Earth and Sun

$B(v, T_s)$: Blackbody (Radiance) [W/cm²/ (cm⁻¹)/sr]

[Solar]SolarAngle : Solid angle through which the sun is seen from the observer (sr)

[Solar]TSun: Temperature of the Sun's surface (K)

3.3.5.2 Radiative options in [SPn]

[SPn]source : type of source at which the instrument is looking at. It can be 'none', 'planet' (or 'earth', 'mars', 'venus'), and 'sun'.

[SPn]NeglectThermalSource: indicates if the thermal emission source in the atmospheric layers are neglected, no test is performed to check if it is meaningful to neglect the thermal emission source.

[SPn]NeglectThermalReflection: indicates if the thermal emission from the atmospheric layers reflected on the surface is neglected, no test is performed to check if it is meaningful to neglect the thermal source.

[SPn]IncludeSolarSource: indicates if the solar source are included (reflection on surface).

[SPn]AtmosphereEmission: Function used to derive the emission from the atmosphere.

Padel : Pade with 1 parameter (default)
Pade2 : Pade with 2 parameters
Taufct : More complex function considering the full dependence of $B(\lambda)$ with tau
Bm : $B(\lambda)$ at the average temperature

[SPn]LambertSurfAtmEmi: Indicates if the surface is Lambertian for the reflection of the Atmospheric Emission

3.3.5.3 Radiative options in [SPn_FENn]

[SPn_FENn]Ts : temperature of the source (K). If a number is given, it represents the temperature. If the option 'zpt' is chosen, Ts will be either given in the ZPT file, if not T of the lowest altitude in the ZPT file will be considered.

[SPn_FENn]Emissivity: gives the emissivity value for the surface (constant value). It will not be used if a File is given ([SPn_FENn]EmissivityFile option).

[SPn_FENn]EmissivityFile: name of the file containing the emissivity. The file must contain 2 columns:

Col 1: Wavenumber (cm-1)
Col 2: emissivity

The file must be located in the [dirPlanet] directory.

[SPn_FENn]Albedo: Albedo of the planet (constant value). If not given, Albedo = 1.0 - Emissivity (including the wavelength dependence if given for Emissivity). Only constant value is valid for the moment.

[SPn_FENn]NeglectThermalSource: indicates if the thermal emission source in the atmospheric layers are neglected, no test is performed to check if it is meaningful to neglect the thermal emission source. Supersedes what is specified in SPn

[SPn_FENn]NeglectThermalReflection: indicates if the thermal emission from the atmospheric layers reflected on the surface is neglected, no

test is performed to check if it is meaningful to neglect the thermal source. Supersedes what is specified in SPn.

[SPn_FENn] IncludeSolarSource: indicates if the solar source are included (reflection on surface). Supersedes what is specified in SPn.

[SPn] AtmosphereEmission: Function used to derive the emission from the atmosphere. Supersedes what is specified in SPn.

Padel : Pade with 1 parameter

Padel2 : Pade with 2 parameters

Taufct : More complex function considering the full dependence of $B(\lambda)$ with tau

Bm : $B(\lambda)$ at the average temperature

[SPn] LambertSurfAtmEmi: Indicates if the surface is Lambertian for the reflection of the Atmospheric Emission. Supersedes what is specified in SPn.

3.3.6 Instrumental functions

The resulting radiance, coming either directly from the RT part of ASIMUT or from LIDORT, is further processed to take into account the instrumental line shape (ILS).

All conventional ILS of Fourier transform spectrometers (Boxcar, Norton Beer strong apodization, ...) are implemented in the program. Specific ILS corresponding to the IMG[55], IASI[1], ACE-FTS[56] instruments are also included. The field of view effect, as well as the distortion of the ILS by off-axis incident light [57], are included. It is also possible to use a user specified ILS, or to correct the ILS with a modulation efficiency and a phase error [58].

Spectra obtained by conventional grating spectrometer are also treated by ASIMUT. Spectra can be simulated either using a wavenumber scale (cm^{-1}) or a wavelength scale (nm).

ASIMUT is also able to simulate spectra recorded using the SOIR instrument [5, 6] on board the Venus Express mission of ESA [4]. The SOIR channel that is part of the SPICAV/SOIR instrument uses a new instrument concept. It combines an echelle grating spectrometer with an Acousto-Optical Tunable Filter for the selection of the appropriate grating orders. This instrument performs solar occultation measurements in the IR region (2.2-4.3 μm) at a spectral resolution of 0.15 cm^{-1} . In such spectrometers, different orders of diffraction combine on the detector. Special care must be taken to accurately simulate the overlapping of successive orders and take into account all instrument characteristics.

3.3.6.1 [Set] options

[Set]Caption: String describing the retrieval (without spaces). If not given, it will be set to the Filaname of the .INP file (without Path and extension).

[Set]zType, [Set]Range: Description of the type of altitude scale on which the Radiative Transfer calculations will be computed. [Set]zType can take the different values:

Range	the 3 values [zmin, zmax, deltaz] (km) are given in [Set]Range.
File	Ascii file with 1 column containing the z values (km). Filename given in [Set]Range.
FITSFIT2	Ascii file with 1 column containing the values (km), but the first line indicates the number of levels. Filename given in [Set]Range.
Values	the values [z1, z2, ...] (km) are given in [Set]Range
Zpt	z scale will be the same as in the zpt file
Horizontal	the output will be given on the Hobs.

[Set]zScale: String describing the retrieval (without spaces). If not given, it will be set to the Filaname of the .INP file (without Path and extension).

[Set]Occ_zmin: Value for tangent height minimum (all files corresponding to a lower altitude are rejected) (km).

[Set]Occ_zmax: Value for tangent height maximum (all files corresponding to a higher altitude are rejected) (km)

[Set]nbSpectra: Number of spectra which are described in the following.

3.3.6.2 [SPn] options

[SPn]InstrumentType: Type of instrument: FTS(includes conventional grating spectrometer) and AOTF

[SPn]Path: Sub directory where the files containing the spectra are located

Full Spectra Directory is:
dirSpectra\Path

Except if Path="full" AND if FileName is a List: the file names given in the list must be given with their full path

[SPn]Filename: Filename for observed spectrum OR name of the file containing a List of filenames

Full path :
Normally the full path to the FileName is:
dirSpectra\Path\FileName,
except if Path="full", in that case the full path to FileName is
dirSpectra\FileName

[SPn]FileType: Type of file

None	No file given, only forward model
Opus	OPUS file
Ascii	Ascii file
Csv	CSV file (with no header, only 2 columns ',' separated)
Opusinfo_csv	CSV file (with header, from OPUSINFO)
IMG_csv	IMG (ascii)
ACE	ACE single spectrum
Iasi	IASI Level 1 spectra (with IASI header)
SOIR_csv	SOIR (csv with header)
SOIR_PDS3	SOIR file - PSA/PDA Level 3 format

[SPn]List: Determines if FileName contains one single measurement file or a list of files. This option is only valid for FileType=Ascii, opus, img_csv, ACE, Soir_csv and Soir_PDS3.

No, 0	Not a list (default)
Yes, 1	The FileName parameters points to a list of files that will be treated sequentially and independently
Occ, 2	The FileName parameters points to a list of files that will be treated simultaneously, as an occultation set
Only valid for	
all, 3	The FileName parameters points to a list of files that will be treated simultaneously

Differences between List / Occ / Onion Peeling:

If [Run]OnionPeeling=no
If **List=yes** (or 1) : all the files are analysed sequentially, on an independent manner. This is an easy way to analyse a series of similar files, with identical fit or simulation parameters.

If **List=occ** (or 2): all files listed are part of one single observation (one single solar occultation) and are all fitted simultaneously. !! Only valid for an occultation set: the altitudes of the tangent heights will define the RT altitude scale that will replace the **[Set]zScale** (except for the altitudes above the highest tangent altitude).

If [Run]OnionPeeling=yes
If **List=yes**: all the files are analysed sequentially, but using the onion peeling method. The spectrum corresponding to the highest altitude is analysed first, then its results are used as input for the analysis of the second one, etc.

[SPn]DataType: Observed spectrum is either a radiance or a transmittance

[SPn]DataXUnit: Units of the X scale of the spectrum. Can be
cm-1 Wavenumber
nm Wavelength

For AOTF instrument only 'cm-1' is possible

[SPn]Date, [SPn]Time: date and time of observation.

[SPn]Lon, [SPn]Lat: Longitude (-360<Lon<360) and Latitude (-90<Lat<90) of observation.

[SPn]FenList: Index to the selected windows for spectrum SPn.
Example: In case of [1 2 25]: Windows SP1_FEN1, SP1_FEN2 and SP1_FEN25 will be selected.

Several parameters describe the instrument. Some are relevant only for FTS: **[SPn]fov, tilt, focal, mirrorsize, iris**.
Others are common to all instruments' types:

[SPn]ILS, [SPn]FileILS: Instrumental Line Shape, and associated file containing the information if required.

Boxcar
Triangle
Trapeze
Happ Genzel
Blackmann Harris 3P
Blackmann Harris 4P
Norton Beer Weak
Norton Beer Medium
Norton Beer Strong
Filename (ils function in the spectrum space)
Filename for eap/eah correction (3 columns: opd, modulation, phase; if only 2 columns: no phase)
Gaussian
ILS fct of ACE
No ILS applied

[SPn]Resolution: Instrument or final resolution. Can be either a Value or a code to indicate which File Format to read. For the moment

the only option is SOIRPDS, to read RESOL_xxx.TAB kind of files from the SOIR PDS.

Units (cm-1 or nm) specified by *DataXUnit* (! The Bruker definition for the resolution of FTS is used=0.9/MOPD).

[SPn]FileResolution: Name of the file containing the information relative for the determination of the resolution (For the moment: only for SOIR format from the PDS Archive, *[SPn]Resolution=SOIRPDS*).

[SPn]HighResolution: High resolution used for the calculation of the Optical Depths. Units (cm-1 or nm) specified by *DataXUnit*
! Normally should not be imposed: ASIMUT determines the optimal resolution based on P and T in each atmospheric layers
To be used only for tests

[SPn]Step: Final step(for simulation only). Units (cm-1 or nm) specified by *DataXUnit*

[SPn]oversampling: Oversampling. If the value is ≤ 0.0 , the oversampling will be calculated in an automatic way by ASIMUT from the 'optimized' resolution.

[SPn]Zerofilling: Zero filling factor (only 2,4, and 8 are possible). The number of points of the spectrum is augmented by a factor 'zerofilling' using the well-known technique in Fourier analysis.

[SPn]Filter: File containing a Filter bandpass (given in transmittance) to apply to the radiance.

[SPn]noiseFile: Name of the file containing the noise.
2 columns : x-absolute noise (same unit as the spectrum), where x= wavenb or wavelength, depending on *DataXUnit*
The file is in *[Directories]Instrument*.

3.3.6.3 OPUS instrument

The OPUS file format is used by BRUKER FTS. See documentation of the FTS to have more information on this format.

The Opus_csv option reads OPUS file converted into csv with a header containing the following information:

```
Zenith angle,153.864042
MOPD,250.000000
Apodization,BX
Date,20041009
Hour,7
Min,18
Sec,8
Fov,0.110685
Iris,0.850000
YMax,2.920782
NbPoints,530994
```

followed by 2 columns with Wavenumber (cm^{-1}) – measured spectrum.

Only the 'opus' option can be used with the List option. The .LST file must then contain 2 columns: Name of the Opus file – Name of the corresponding ZPT file.

3.3.6.4 IMG instrument

The file is in Ascii (csv, comma delimited). It contains a header:

```
"DATE","27-12-1996 12:13:53"  
"LATITUDE"," -4.8 deg"  
"LONGITUDE"," -23.7 deg"  
"SURFACE","Sea"  
"ALTITUDE"," 600.0 m"  
"ECMWF","era_1996122702_38135.annc"  
"UNITS","cm-1","W/cm2/Sr/cm-1"
```

followed by 2 columns with Wavenumber (cm⁻¹) - Radiance
The "ECMWF" line of the header indicates which zpt file to read. The file must exist.

Option List is possible with this instrument. The .LST file must then contain 2 columns: Name of the Opus file - Name of the corresponding ZPT file.

3.3.6.5 ACE instrument

The file containing the ACE spectra are in ascii, 2 columns, Wavenumber (cm⁻¹) - Transmittance.

ACE spectra can be used with the LIST option. The .LST file must then contain the following information

```
Path  
BaseName  
Index of spectrum 1 Tangent height of spectrum 1  
Index of spectrum 2 Tangent height of spectrum 2  
...  
Index of spectrum n Tangent height of spectrum n
```

The name of the file containing each of the spectra in the list will then be reconstructed using: Path/BaseName_spectNNN.dat, where NNN is the index of the spectrum.

The same .LST file can be used with the 'ACE_occ' option. All spectra listed are then analysed simultaneously.

3.3.6.6 IASI instrument

The IASI files are Ascii (csv, comma delimited). They contain a header:

```
"MISSION","IASI/METOP"  
"DATE","01-07-2007 11:31:14"  
"LATITUDE",-34.644520,"deg (North)"  
"LONGITUDE",151.112400,"deg (East)"  
"ZENITH",31.720000,"deg"  
"AZIMUTH",77.520000,"deg"  
"SURFACE","unknown"  
"ALTITUDE",0,"km"  
"HOBS",833800.000000,"km"  
"DISSTANCE",30.350771,"km"  
"ZPT","ZPT_2007070121_110662.txt"  
"UNITS","cm-1","W/cm2/Sr/cm-1"
```

followed by 2 columns with Wavenumber (cm⁻¹) - Radiance
The "ZPT" line of the header indicates which zpt file to read. The file must exist.

Option List is possible with this instrument. The .LST file must then contain only 1 column with the Name of the IASI file.

3.3.6.7 SOIR instrument

The parameters describing the instrument and spectrum or list of spectra are described, as well as the format of the different files needed.

[SPn]aotfFrequency: AOTF frequency. This information is also read directly from SOIR_CSV and SOIR_PDS type of files.

[SPn]aotfCentralWnb: AOTF central wavenumber (corresponding to the aotfFrequency). If aotfCentralWnb is given, it has the precedence over aotfFrequency

At least one of the two parameters must be present except when using the SOIR_CSV and SOIR_PDS3 type of files.

[SPn]CentralOrder: AOTF central Order (only used when reading RESOL file if no Data are specified). Normally this value is read from the header of the SOIR files.

[SPn]aotfNbPixel: Number of pixels on the detector.

[SPn]NumberOrders: Number of adjacent orders included in the SOIR type spectra.

[SPn]aotfbinningOpt: Binning option on the detector. This will impose also the Maximum number of bins (for binning 3,4: 8 bins; for binning 12,16: 2 bins).

[SPn]aotfbinNb: Bin number to consider. For example using the 'binning 12' option, two bins are recorded simultaneously.

[SPn]AOTFFunction: Type of AOTF function to use. It can be
Sinc2 Simple sinc2,
5Sinc2 Sum of 5 sinc2,
File From ASCII File (2 columns: nu-aotf)
filePDS From PDS file

When a file ('file', or 'filePDS' options) is required, its name is given in **[SPn]FileAotfFilter**.

[SPn]FileAotfFilter: File containing the AOTF band pass characterization, either as parameterization or the complete function.

FileAotfFilter format - Type FILE

One file per bin
AOTF filter bandpass function
11 : param_1 p1 p2 p3
12 : param_2 p1 p2 p3
...
ln : param_n p1 p2 p3

Values of the parameters are calculated using a polynomial relation
= p1+p2 x aotf_nucentral + p3 x aotf_nucentral^2, using maximum 3
coefficients (p1..p3)

ex:

```
%Created by get_all_work_miniscan.m.  
%Date: 12-Sep-2007 10:23:23.  
%Bin 1  
%Binning: 12.  
23.7537542783189 -0.000164628838726562
```

```

11.3345861384054  0.00251115825898408
150.41787923001 -0.0302391265256812
104.843380735229 -0.0219931132766153
134.798482794379 -0.0285662006934839
      0          0
-12.1817220123877 -0.00174231790834488
21.5151538373025  0.00129500896606321
-83.5570832000095  0.00754897676743825
36.8957781658948  0.00461531397182787
0.962610146433146 6.63944157657772e-006
0.0789788682854946 4.12654538910255e-006
-0.281796550292296 9.48605838175013e-005
-0.0353052533554235 2.67081772054492e-005
0.232077580267617 -4.72433590075699e-005

```

Which defines the 15 parameters needed for the construction of the aotf function consisting of the sum of 5 sinc² (AOTF_5SINC2) (5 first lines = fwhm of the 5 sinc, then 5 lines for the displacement relative to the center of the Aotf fct, finally 5 lines for the intensity factor). For each parameter, 2 coefficients are given, defining a linear variation of the coefficients with Wn
For ex, the FWHM of the first sinc wil be given by =
23.7537542783189 + -0.000164628838726562 x aotf_nuCentral

FileAotfFilter - Type FILEPDS

Same format as the one given in the PDS ESA Archive. One file per binning and bin.

```

AOTF filter bandpass function
Order1, bin1, WaveNb start, WaveNb end, pas, value 1 ... value N
Order1, bin2, WaveNb start, WaveNb end, pas, pas, value 1 ... value N
Order2, bin1, WaveNb start, WaveNb end, pas, pas, value 1 ... value N
Order2, bin2, WaveNb start, WaveNb end, pas, pas, value 1 ... value N

```

```

ex: AOTF_TF_BINNING12.TAB
101, 1, -100.0, 100.0, 0.1, 1.4095e-03, 1.3764e-03, 1.3456e-03, ....
101, 2, -100.0, 100.0, 0.1, 2.0169e-03, 2.0290e-03, 2.0436e-03, ....

```

[SPn] FileTuningFct: File containing the AOTF tuning functions (freq-wavenb): Wavenb=a +bxfreq+cxfreq².

FileTuningWavenb format

The format is the same as the PSA/PDS one:

```

Calibration (freq AOTF - wavenb) [TuningFct]
  // file format:
  // cf ARCHIVE ... \CALIB\AOTF_F_WN.TAB
  // with explanation of format in AOTF_F_WN.LBL
"F->WN", binning, bin number, Param 1, Param 2, Param 3, Param 4, Param 5
"WN->F", binning, bin number, Param 1, Param 2, Param 3, Param 4, Param 5

```

```

ex: AOTF_F_WN.TAB
"F->WN", 3, 1, 0.0000e+00, 0.0000e+00, 2.4852e-07, 1.4577e-01, 3.5373e+02
"F->WN", 3, 2, 0.0000e+00, 0.0000e+00, 2.4935e-07, 1.4563e-01, 3.5526e+02

```

[SPn] FileResolution: File containing the resolution (cm-1) of the SOIR instrument.

FileResolution format

The format is the same as the PSA/PDS one:

```

Resolution (cm-1)
  // file format:
  // cf ARCHIVE ... \CALIB\RESOL_BINNING12.TAB
  // with explanation of format in AOTF_F_WN.LBL

```

```
Order, resolution in bin 1, Resolution in bin 2
```

```
ex: RESOL_BINNING12.TAB
    101, 1.0956e-01, 1.0749e-01
    102, 1.1058e-01, 1.0855e-01
    103, 1.1161e-01, 1.0961e-01
    104, 1.1264e-01, 1.1067e-01
```

[SPn] FileCalibWavenb: File containing the parameters for calibrating the wavenumber scale. The parameters needed for the conversion pixel-wavenumber, and wavenumber-pixel are given. Normally these parameters are also found in the headers of the SOIR_PDS and SOIR_CSV files. In that case, the later have priority. ASIMUT will only read the "PIX->WN" line and will compute the parameters for the Wn->PIX conversion from it.

FileCalibWavenb format

```
Calibration (pixel-wavenb)
    // file format:
    // cf ARCHIVE ... \CALIB\PIX_WN.TAB
    // with explanation of format in PIX_WN.LBL
    //
"PIX->WN",binning,bin number, Param 1, Param 2, ... Param 5
"WN->PIX",binning,bin number, Param 1, Param 2, ... Param 5

ex: PIX_WN.TAB
"PIX->WN",12,1,0.0000,0.0000, 4.8623e-008, 5.8565e-004, 2.2348e+001
"PIX->WN",12,2,0.0000,0.0000, 6.1550e-008, 5.8211e-004, 2.2348e+001
"WN->PIX",12,1,0.0000,0.0000, -2.2366e+002, 1.1703e+004,-1.4983e+005
"WN->PIX",12,2,0.0000,0.0000, -2.8226e+002, 1.4331e+004,-1.7931e+005
```

SOIR csv and SOIR csv occ formats

The files are in Ascii CSV format (comma separated). Each file starts with a header:

```
date, 2006-05-20
TangHeight, 87.478120, 87.442389
DistVenus, 1211.655717, 1211.655717
Angle, 122.303598, 122.304126
AOTFFrequency, 25709.000000, 25709.000000
Lon, 21.693378, 21.697435
Lat, 82.686342, 82.686230
LST, 17.193611, 17.193333
PixWn_bin1, 0000000000000000, 4.396840000000000e-010, -1.801220000000000e-007, 6.232410000000000e-004, 2.234470000000000e+001
PixWn_bin2, 0000000000000000, -5.209850000000000e-010, 3.006560000000000e-007, 5.479830000000000e-004, 2.234840000000000e+001
tiltAngle, 0.199909, 0.041842
Hslit, 2.319432, 2.319432
spdVEXSun, -8.688690, -8.688690
spdVenusSun, -0.129398, -0.129398
spdVenusOrbit, -2.741840, -2.740380
errorAlt, 0.113087, 0.113089
```

followed by $N \times 2$ columns, where N is the number of bins corresponding to the 'binning' option of the observation. The N first columns are the wavenumber (cm^{-1}) and the following N ones contain the transmittance.

The MATLAB macro *ConvertPDS_ASCII.m* used to convert PDS file to CSV and to create some additional files (such as the LST file, giving the

list of all files converted) is also provided in the ASIMUT\MATLAB subdirectory.

The difference between SOIR_csv and SOIR_csv_occ is that all files are analysed individually in the first case, or all together in the second. (See also the examples provided)

SOIR PDS3 format

These files correspond to the PSA ESA archive Level 3 data. Their format is explained in the Archive in details.

One file contains all the data concerning one occultation observation. One line gives all the information and data on one spectrum (obtained on one bin). The different information are the : date, time, bin, binning, alt, pointing angle (degree), distance to Venus , tilted angle of the slit , slit height, latitude, longitude, local solar time (lst), speed of VEX wrt Sun, speed of Venus wrt Sun, speed of Venus on Orbit, error on the Altitude, AOTF frequency, integration time, number of accumulation, pixel-wavenumber conversion coefficients (5 values maximum),Transmittance (320 values), error on Transmittance (320 values), followed by 16 parameters describing the housekeeping.

3.3.6.8 [SPn_FENx] options

[SPn_FENx]Pass: gives a PASS number to the window. All window having the same PASS number will be fitted simultaneously, otherwise the fit is sequential. For example if [SP1_FEN2]Pass = 1, [SP2_FEN44]Pass = 1 and [SP1_FEN1]Pass = 2 : the spectra corresponding to the window 2 of spectrum 1 and window 44 of spectrum 2 will be fitted together, then the spectrum of window 1 of spectrum 1 will be fitted.

A series of parameters described the wavenumber (wavelength) limits of the window. The unit in which those limits are expressed depend on the [SPn]DataXUnit flag.

[SPn_FENx]wavemin, wavemax, waveminplot, wavemaxplot

[SPn_FENx]NbdeWeight, [SPn_FENx]deWeightedLimits: It is possible to indicate regions inside the limits which will be fitted with a different weight as the rest of the window. This is usefull to remove from the fit a bad pixel or a known saturated line. The two parameters indicate the number of such deweigthed regions and their limits

[SPn_FENx]shift: constant shift to apply on the wavenumber scale of the spectrum.

[SPn_FENx]Calibration : for FTS, multiplicative correction factor applied on the wavenumber scale. The value given is indeed the value of the multiplicative factor -1.0. All wavenumbers will be multiplied by the factor =(1.0+[SPn_FENx]Calibration).

A series of parameters described the instrument. There supersede the values given in [SPn] (fov, tilt, focal, mirrorsize, iris, oversampling)

[SPn_FENx]noiseType: Type of noise on the observations. Can be :

Cst	Constant value (either [SPn_FENx]snr or [SPn_FENx]snrabs)
fromSpectrumFile	Noise is read from the Spectrum file (only possible now for SOIR type of spectra)
fromFile	From an external file, whose name is specified under [SPn]noiseFile.

[SPn_FENx]snr, snrabs: Signal to noise ratio and absolute noise level. At least one of them must be given (used in the Rodgers formalism). The Absolute snrabs must be given in the same units as the spectrum.

[SPn_FENx]Seddiag: Indicates if Se (noise on measurement) is diagonal. If not diagonal, the non-diagonal elements of the Se matrix are calculated by applying a Gaussian function with a width equals to the resolution of the instrument.

The following parameters describe which molecules (LBL or cross sections), continua or aerosols are included in the window.

[SPn_FENx]molecules: Molecules (LP type) to be included (Line-by-line calculations).

Maximum number of Molecules that can be included in one run : 129
(=number of recognized isotopologues).

[SPn_FENx]FitMolecules: For each molecule, indicates type of fit:

Value is < 0.0 Fit only the column
Value is = 0.0 No fit, only simulation
Value is >0.0 Fit as described in [xxxLP] description

The decimal part of the number is the covariance for the fit (if no Sa file is given).

[SPn_FENx]aPrioriMolecules: For each molecule, indicates the type of a priori that will be used in the Rodgers formalism:

Model From the atmosphere file
Previous From previous fit
ATMname From previous fit of ATMname molecule.

The last option is useful when fitting separately different isotopologues of the same molecule: for example, first (in PASS=1) fit only the main isotopologue O3 666, then in a second PASS where isotopologue O3 686 is fitted, indicating aPrioriMolecules=[O3] will strat the fit of O3 686 from the results of the fit of O3 done in the previous pass.

[SPn_FENx]ODLimit: Limit of Optical Depth under which a line will not be considered.

[SPn_FENx]LBLfct: Type of function used for the LBL calculation, using either a constant or variable step.

[SPn_FENx]CrossSections: cross sections (XS type) to be included .
Maximum number of cross sections that can be included in one run : 20

[SPn_FENx]FitCrossSections: For each cross section, indicates type of fit:

Value is < 0.0 Fit only the column
Value is = 0.0 No fit, only simulation
Value is >0.0 Fit as described in [xxxXS] description

The decimal part of the number is the covariance for the fit (if no Sa file is given).

[SPn_FENx]aPrioriCrossSections: For each cross section, indicates the type of a priori that will be used in the Rodgers formalism:

Model From the atmosphere file
Previous From previous fit
ATMname From previous fit of ATMname molecule.

[SPn_FENx]Aerosols: Aerosols (AER type) to be included (Line-by-line calculations).

Maximum number of Aerosols that can be included in one run : 10

[SPn_FENx]FitAerosols: For each Aerosol, indicates type of fit:

Value is < 0.0 Fit only the column
Value is = 0.0 No fit, only simulation
Value is >0.0 Fit as described in [xxxAER] description

The decimal part of the number is the covariance for the fit (if no Sa file is given).

[SPn_FENx]aPrioriAerosols: For each Aerosol, indicates the type of a priori that will be used in the Rodgers formalism:

Model From the atmosphere file

Previous From previous fit
ATMname From previous fit of ATMname molecule.

[SPn_FENx]Continua: indicates which continua will be included.
Maximum number of Aerosols that can be included in one run : 5

[SPn_FENx]Rayleigh: indicates if Rayleigh diffusion will be included.

A series of parameters describe the 'background' of the spectrum.

[SPn_FENx]BaselineCst: indicates the Baseline constant term. If set to '-1', the value is automatically determined from the measurement file.

[SPn_FENx]BackgroundWavenumbers, BackgroundValues: defines the background, size of the two vectors must be the same.
A series of parameters describe the radiative properties to be used in the window.

[SPn_FENn]Ts : temperature of the source (K). If a number is given, it represents the temperature. If the option 'zpt' is chosen, Ts will be either given in the ZPT file, if not T of the lowest altitude in the ZPT file will be considered.

[SPn_FENn]Emissivity: gives the emissivity value for the surface (constant value). It will not be used if a File is given (**[SPn_FENn]EmissivityFile** option).

[SPn_FENn]EmissivityFile: name of the file containing the emissivity.
The file must contain 2 columns:
Col 1: Wavenumber (cm-1)
Col 2: emissivity
The file must be located in the **[dirPlanet]** directory.

4. Onion peeling method

The onion peeling method was implemented to coherently treat a series of spectra recorded during one occultation. In this method, one starts the analysis in the uppermost layer, i.e. with the first spectrum containing absorption structures due to the constituents of the atmosphere, deriving concentrations in that layer, and progressively goes deeper into the atmosphere taking into account the results from the layers above. Vertical profiles of several key species of the Venusian atmosphere have been obtained by applying this technique, as will be demonstrated hereafter.

For the sake of clarity, we will consider in the following that the second right term in the general equation describing the radiative transfer through the atmosphere

$$I(\nu) = I_0(\nu) e^{-\tau(\nu, 0, s_{obs})} + \int_0^{s_{obs}} B(\nu, T(s)) \alpha(\nu, s) e^{-\tau(\nu, 0, s)} ds \quad (4.1)$$

is negligible.

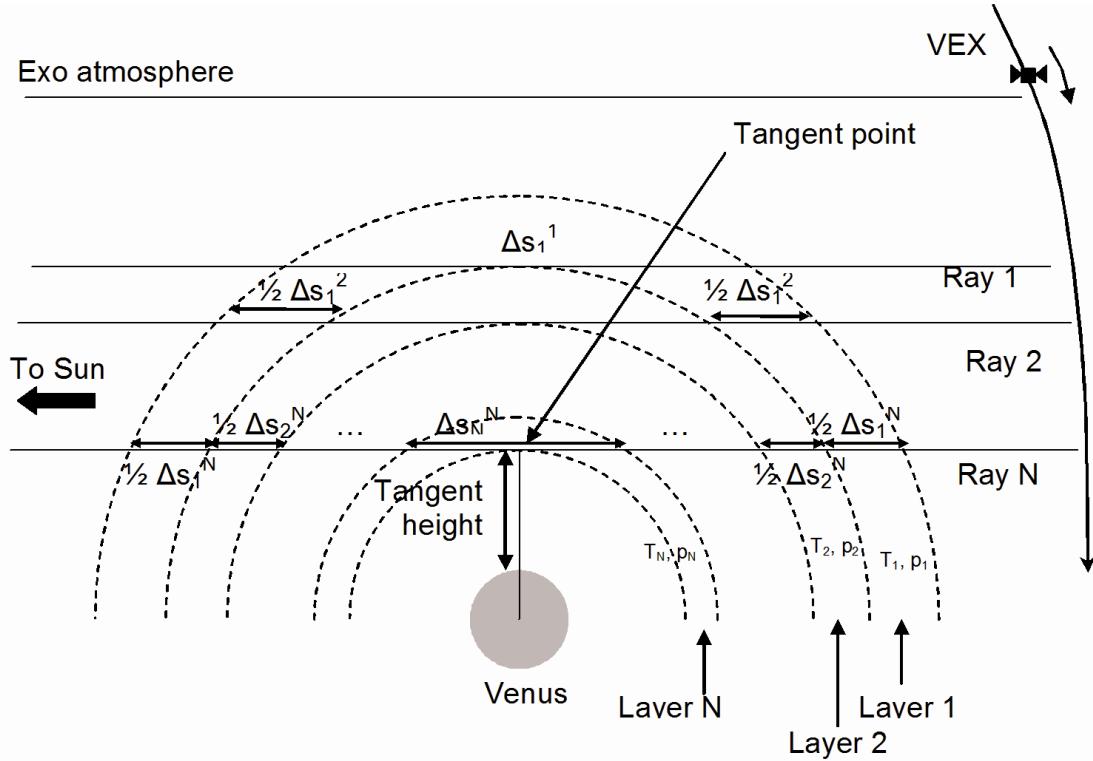


Figure 5: Geometry of solar occultation measurements and definition of the onion peeling method.

The observed transmittance Tr_1 corresponding to Ray 1 passing through the uppermost layer (Layer 1 in Figure 5) is then given by:

$$Tr_1(\nu) = \exp[-\alpha_1(\nu)\Delta s_1^1] \quad (4.2)$$

where $\alpha_1(\nu)$ stands for $\alpha(T_1, P_1, \nu)$ and Δs_1^1 is the length of the ray path in Layer 1 obtained by the raytracing procedure. In this expression, the only unknowns are the concentration N_i of each species and the aerosol loading in Layer 1, which are

retrieved from the analysis of this first layer. Transmittance observed for Ray 2 will result from the combination of the absorption of light in Layer 1 (Δs_1^2) and Layer 2 (Δs_2^2). If we moreover consider the atmosphere as spherical and homogeneous, we can further write:

$$Tr_2(\nu) = \exp[-\alpha_2(\nu)\Delta s_2^2 - \alpha_1(\nu)\Delta s_1^2] \quad (4.3)$$

in which the only unknowns are the concentration N_i of each species and the aerosol loading in Layer 2. By going down progressively, the vertical profiles of the interacting species can be derived.

The onion peeling method [59] has been implemented in ASIMUT for solar occultation measurements. This method derives vertical information on the retrieved species, starting at the most distant (top) layer and going down toward the lowest, taking into account the information obtained the way down. Spectra recorded by the SOIR instrument [5, 6] on board the Venus Express mission of ESA [4] have successfully been investigated with ASIMUT [12, 15]. The ACE-FTS instrument [56] also measures atmospheric absorptions by solar occultation, and the same method can be used in this case.

4.1 Onion peeling options

[OnionPeeling]OnionPeeling: if =1, the onion peeling method will be used.

[Set]Occ_zmin: Value for tangent height minimum (all files corresponding to a lower altitude are rejected) (km).

[Set]Occ_zmax: Value for tangent height maximum (all files corresponding to a higher altitude are rejected) (km)

5. Retrieval module

Two main objectives in the development of ASIMUT have been to use the Optimal Estimation Method (OEM) [7] coupled to the analytical calculation of the Jacobians, and to allow simultaneous retrieval from a series of spectra obtained under different observation geometries, with the possibility to combine spectra recorded by different instruments.

5.1 General description

The general forward radiative transfer equation can be written as:

$$\mathbf{y} = \mathbf{f}(\mathbf{x}, \mathbf{b}) + \varepsilon \quad (5.1)$$

where \mathbf{y} is the measurement vector (the measured radiance), \mathbf{x} is the state vector (the vertical profiles to be retrieved), \mathbf{b} represents the additional parameters used by the forward model, \mathbf{f} . The forward function \mathbf{f} describes the complete physics of the measurement, including the description of the instrument. In the case of a moderately nonlinear problem, the best estimate $\hat{\mathbf{x}}$ of the solution of Eq. (5.1) is found by solving iteratively

$$\mathbf{x}_{i+1} = \mathbf{x}_a + \mathbf{G}_i [\mathbf{y} - \mathbf{f}(\mathbf{x}_i) + \mathbf{K}_i(\mathbf{x}_i - \mathbf{x}_a)] \quad (5.2)$$

where \mathbf{x}_a is the a priori constraint. The Jacobian matrix \mathbf{K} is defined by the following expression:

$$\mathbf{K} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \quad (5.3)$$

and the Gain matrix \mathbf{G} is defined by:

$$\mathbf{G}_i = (\mathbf{S}_a^{-1} + \mathbf{K}_i^T \mathbf{S}_e^{-1} \mathbf{K}_i)^{-1} \mathbf{K}_i^T \mathbf{S}_e^{-1} \quad (5.4)$$

where \mathbf{S}_e the error covariance matrix of the measurements, and \mathbf{S}_a the a priori covariance matrix. \mathbf{S}_e is chosen to be diagonal, characterizing all the sources of systematic and random errors on the measured radiance.

The convergence is attained when the 2 following conditions are met:

$$\begin{aligned} d_i^2 &= (\mathbf{x}_i - \mathbf{x}_{i+1})^T \mathbf{S}^{-1} (\mathbf{x}_i - \mathbf{x}_{i+1}) \ll n \\ d_i^2 &= (\mathbf{f}(\mathbf{x}_{i+1}) - \mathbf{f}(\mathbf{x}_i))^T \mathbf{S}_{\delta y}^{-1} (\mathbf{f}(\mathbf{x}_{i+1}) - \mathbf{f}(\mathbf{x}_i)) \ll m \end{aligned} \quad (5.5)$$

with n the number of parameters to retrieve, m is the number of observed points, and

$$\delta \mathbf{y} = \mathbf{y} - \mathbf{f}(\hat{\mathbf{x}}) \quad (5.6)$$

$$\mathbf{S}_{\delta y} = \mathbf{S}_e (\mathbf{K} \mathbf{S}_a \mathbf{K}^T)^{-1} \mathbf{S}_e \quad (5.7)$$

The first expression expresses the convergence condition on the fitted parameters, and the second on the difference between observed and simulated spectra.

The matrix \mathbf{S}

$$\mathbf{S} = (\mathbf{S}_a^{-1} + \mathbf{K}_i^T \mathbf{S}_e^{-1} \mathbf{K}_i)^{-1} \quad (5.8)$$

represents the total error covariance on the retrieved parameters, which can be decomposed into two contributions: the smoothing error, which accounts for the sensitivity of the measurements/forward model to the variable to be retrieved (i.e. the measurement/forward model system does not allow perfectly reproducing the true atmosphere, but a smoothed value of it), and the measurement error. Their covariance can be expressed as:

$$\mathbf{S}_{smoothing} = (\mathbf{A} - \mathbf{I})\mathbf{S}_a(\mathbf{A} - \mathbf{I})^{-1} \quad (5.9)$$

$$\mathbf{S}_{measur} = \mathbf{G}\mathbf{S}_e\mathbf{G}^T \quad (5.10)$$

where $\mathbf{G} = \frac{\partial \hat{\mathbf{x}}}{\partial \mathbf{y}}$ is the gain matrix and $\mathbf{A} = \frac{\partial \hat{\mathbf{x}}}{\partial \mathbf{x}} = \mathbf{G}\mathbf{K}$ is the averaging kernel matrix. \mathbf{I} is the identity matrix.

The basic implementation of the Rodgers algorithm in ASIMUT makes use of approximate weighting functions. The main objective is to save computation time. In this approach, the Jacobians are calculated for the first guess (\mathbf{K}_0), and are not updated at each iteration:

$$\mathbf{K}_i = \mathbf{K}_{i+1} = \dots = \mathbf{K}_0 \quad (5.11)$$

This also implies that:

$$\mathbf{G}_i = \mathbf{G}_{i+1} = \dots = \mathbf{G}_0 \quad (5.12)$$

and

$$\mathbf{x}_{i+1} = \mathbf{x}_a + \mathbf{G}_0[\mathbf{y} - \mathbf{f}(\mathbf{x}_i) + \mathbf{K}_0(\mathbf{x}_i - \mathbf{x}_a)] \quad (5.13)$$

In that case the solution converge to \mathbf{x}_∞ , satisfying the equation

$$\mathbf{x}_\infty = \mathbf{x}_a + \mathbf{G}_0[\mathbf{y} - \mathbf{f}(\mathbf{x}_\infty) + \mathbf{K}_0(\mathbf{x}_\infty - \mathbf{x}_a)] \quad (5.14)$$

Convergence will be defined by

$$\mathbf{x}_{i+1} - \mathbf{x}_\infty = \mathbf{G}_0[\mathbf{K}_0 - \mathbf{K}_\infty](\mathbf{x}_i - \mathbf{x}_\infty) + O(\mathbf{x}_i - \mathbf{x}_\infty)^2 \quad (5.15)$$

This approach is correct if the forward model is linear between \mathbf{x} and \mathbf{x}_∞ , which is a valid hypothesis for moderately nonlinear problems. In such cases, it is only necessary to calculate \mathbf{K}_0 and \mathbf{K}_∞ at the first and last iteration respectively and to use \mathbf{K}_∞ in all equations defining the error on the fit.

ASIMUT however offers the possibility to update the \mathbf{K} at each iteration, when the problem cannot be considered linear enough. Tikonov regularization method is also implemented.

5.1.1 [Run] options

[Run]Rodgers: imposes a constraint on the fitted x when these are <0.0.

Nothing	normal way to use Rodgers. No constraint. Fitted x can be <0.0. Note that in ASIMUT, the concentrations (not their logarithm) are fitted. X values <0.0 are thus meaningless. Also note that x<0.0 leads to an error if using (V)LIDORT.
Zero	any x values <0.0 are set to 0.0
Apriori	set the negative fitted x values to their apriori xa values

[Run]Derivatives: Jacobians are supposed to vary smoothly wrt the fitted parameters. In that case they are calculated only once at the beginning of the retrieval (and once at the end to calculate the error). If this hypothesis is not true, set [Run]Derivatives = each,

and the Jacobians will be recalculated at each step of the iterative Rodgers algorithm.

[Run]OptimizeRodgers: With that option the Rodgers algorithm will be the following: use the approximate Jacobians (K0 calculated once at the first iteration), after convergence, recalculate the new K0 Jacobians and redo a full iterative process. Repeat until convergence is achieved on both criteria x and f. This is a mix between the approximate Jacobians and the full Rodgers (K updated at each iteration).

[Run]InterpolMethod: When fitting a vertical profile (see XX), the user can specify levels on which the fit will be done. These levels must not be the same as the ones used for the Radiative Transfer calculations. The [Run]InterpolMethod flag indicates how to extrapolate the fitted factors to all levels. If set to 0, the extrapolated fitted factors on layers above or below the fitted levels are set to the top and bottom fitted values respectively. If set to 1, all factors in layers above and below the fitted levels are set to 1.0 (=not fitted, use the a priori profile).

5.2 Determination of the Jacobians

All Jacobians used by the retrieval module are computed analytically during the forward model calculations. The computation is based on the separability of the functions representing on the one hand the radiative forward model $\mathbf{F}_R(\mathbf{x}, \mathbf{b}_R)$ and on the other hand the instrument $\mathbf{I}(\mathbf{b}_i)$:

$$\mathbf{y} = \mathbf{f}(\mathbf{x}, \mathbf{b}) + \varepsilon = \mathbf{I}(\mathbf{b}_i) \otimes \mathbf{F}_R(\mathbf{x}, \mathbf{b}_R) + \varepsilon \quad (5.16)$$

in which \mathbf{b}_R and \mathbf{b}_i are respectively the model parameters needed for the radiative transfer part and the instrument description respectively. Note that, implicitly, Eq. (5.16) means that it is not possible to retrieve instrumental parameters, such as modulation or phase error, with ASIMUT.

From Eq. (3.1), the following derivatives can be deduced:

$$\frac{\partial \mathbf{F}_R}{\partial T_0} = \frac{\partial I_0(T_0)}{\partial T_0} e^{-\tau(\nu, 0, s)}; \quad \frac{\partial \mathbf{F}_R}{\partial n_{ik}} = \frac{\partial \mathbf{F}_R}{\partial \tau_k} \frac{\partial \tau_k}{\partial n_{ik}} \quad (5.17)$$

The $\frac{\partial \mathbf{F}_R}{\partial \tau_L}$ derivatives are calculated along the path using relations similar to Eq. (5.47). The $\frac{\partial \tau_L}{\partial n_{il}}$ are computed for each species at each altitudes taking into account

the different contributing terms (LBL term, cross section, continua). For H₂O, O₂, N₂, and CO₂, the derivatives of the Rayleigh scattering are included. Moreover, the O₂ continua expressions depends on the N₂ density (and vice-versa), and the corresponding cross-terms are also included.

5.3 A priori covariance matrices

The a priori covariance matrices (\mathbf{S}_a) should represent the variability of the gas/aerosol considered. This information is not always available. ASIMUT can manage different types of \mathbf{S}_a matrices: full matrices provided by the user and obtained

with the aid of climatologies, diagonal matrices, diagonal matrices with off-diagonal terms added as Gaussian

$$S_a(i, j) = \sqrt{S_a(i, i)S_a(j, j)} \exp\left(-\left(\frac{z_i - z_j}{l_c}\right)^2\right) \quad (5.18)$$

or exponential functions

$$S_a(i, j) = \sqrt{S_a(i, i)S_a(j, j)} \exp\left(-\frac{|z_i - z_j|}{l_c}\right) \quad (5.19)$$

to account for correlations between the concentrations at different altitudes (z_i is the altitude of level i , l_c is the vertical correlation length).

5.3.1 Covariance options in [xxxLP], [xxxXS], and [xxxAER]

All the following parameters are valid for molecules (LP), cross sections (XS) and aerosols (AER) and indicate how the fit will be done and which covariance will be selected.

[xxxMM] zfit: altitudes on which the fit will be performed. Several options are possible:

```

Zfact      same altitudes as those given in
           [xxxMM]zfact
zRay      same altitudes as those used for the
           Radiative Transfer calculations
           ([Set]zScale
[z1, z2,...] altitudes

```

[xxxMM] SaType: description of the Sa values:

```

None      No Sa given (decimal part of FitMolecules,
          FitCrosssections, or FitAerosols for all diagonal
          Sa values)
Values    Values given in '[xxxMM]Sa'
gauss     Values given in '[xxxMM]Sa' + Gaussian correlation
Expon     Values given in '[xxxMM]Sa' + exponential
          correlation
File      Sa values in file (1st col=altitudes)
DiagFromFile Take only diagonal values of Sa in file
GaussFromFile Diagonal values of Sa in file + Gaussian
               correlation
ExponFromFile Diagonal values of Sa in file + exponential
               correlation

```

[xxxMM] Sa: Sa definition, content depends on [xxxMM]SaType. Can be a FileName or values.

[xxxMM] SaH: Mixing height (km), used with the option 'Gauss' and 'Expon'.

[xxxMM] SaZ: Altitudes (km) at which Sa diagonal values are given in 'Sa'.

[xxxMM] UseTikonov: indicates if the Tikonov regularization method is used.

[xxxMM] TikonovFile: Filename containing the Tikonov matrix which should be a square matrix defined on the same altitude grid as the one given by [xxxMM]zfit.

5.4 Summary of all Variables that can be fitted

Several variables can be fitted: obviously the densities of molecules, cross sections and aerosols (columns and/or profiles), the background signal, the surface temperature. This section gives an overview of all fit possibilities and the calling convention (options in the .INP files).

5.4.1 Molecules, cross sections and aerosols densities

Columns and profiles can be fitted by specifying the following parameters:

```
[SPn_FENx]FitMolecules
[SPn_FENx]FitCrossSections
[SPn_FENx]FitAerosols:
```

For each molecule, cross section or aerosol, indicates type of fit:

Value is < 0.0	Fit only the column
Value is = 0.0	No fit, only simulation
Value is >0.0	Fit as described in [xxxLP], [xxxxXS], or [xxxAER] description

The decimal part of the number is the covariance for the fit (if no Sa file is given).

```
[SPn_FENx]aPrioriMolecules
[SPn_FENx]aPrioriCrossSections
[SPn_FENx]aPrioriAerosols:
```

For each molecule, cross section or aerosol, indicates the type of a priori that will be used in the Rodgers formalism:

Model	From the atmosphere file
Previous	From previous fit
ATMname	From previous fit of ATMname molecule.

The last option is useful when fitting separately different isotopologues of the same molecule: for example, first (in PASS=1) fit only the main isotopologue O3 666, then in a second PASS where isotopologue O3 686 is fitted, indicating *aPrioriMolecules*=[O3] will start the fit of O3 686 from the results of the fit of O3 done in the previous pass.

The Covariance is described in each corresponding [xxxLP], [xxxxXS], or [xxxAER] under the *Sa*, *SaType*, ... parameters.

5.4.2 Variables related to the Spectrum and its windows

Several variables can be fitted: wavenumber shift ([SPn_FENx]FitShift), the surface temperature ([SPn_FENx]FitTs), the signal baseline ([SPn_FENx]FitBaselineCst, FitBaselineX and FitBaselineX2), the albedo ([SPn_FENx]FitAlbedo).

[SPn_FENx]FitShift: Wavenumber shift [SPn_FENx]Shift is fitted. The number given under

[SPn_FENx]FitTs: Fit the surface temperature [SPn_FENx]Ts. Different options are possible:

3.x	Fit of the surface temperature. Take the $[SPn_FENx]Ts$ value previously fitted in another FEN (x= std deviation)
-2	No fit. However, take the $[SPn_FENx]Ts$ value previously fitted in another FEN
-1	Fit simultaneously on another FEN (in which this option will be set to '1.x')
0	No Fit
1.x	Fit the surface temperature (x= std deviation)

$[SPn_FENx]FitBaselineCst$: Fit the baseline constant term. Different options are possible:

3.x	Fit of the baseline constant term. Take the $[SPn_FENx]FitBaselineCst$ value previously fitted in another FEN (x= std deviation)
-2	No fit. However, take the $[SPn_FENx]FitBaselineCst$ value previously fitted in another FEN
-1	Fit simultaneously on another FEN (in which this option will be set to '1.x')
0	No Fit
1.x	Fit the baseline constant term (x= std deviation)

$[SPn_FENx]FitBaselineX$: Fit the baseline 'nu' term. Set to '0' if no fit is requested. Set to '1.x' to fit, with x= std deviation.

$[SPn_FENx]FitBaselineX2$: Fit the baseline 'nu^2' term. Set to '0' if no fit is requested. Set to '1.x' to fit, with x= std deviation.

$[SPn_FENx]FitAlbedo$: Fit the albedo of the surface $[SPn_FENx]albedo$. Different options are possible:

3.x	Fit of the albedo. Take the $[SPn_FENx]albedo$ value previously fitted in another FEN (x= std deviation)
-2	No fit. However, take the $[SPn_FENx]albedo$ value previously fitted in another FEN
-1	Fit simultaneously on another FEN (in which this option will be set to '1.x')
0	No Fit
1.x	Fit the albedo (x= std deviation)

Albedo is fitted only if the geometry allows the reflection of Solar radiation on the surface and is asked by the user.

6. (V)LIDORT description

LIDORT and VLIDORT are capable of simulating plane-parallel and pseudo-spherical atmospheres. The pseudo-spherical approximation means that the solar beam attenuation is calculated in a curved atmosphere. All scattering effects are calculated in a plane-parallel atmosphere.

Both LIDORT and VLIDORT are monochromatic codes. Since ASIMUT operates with a range of wavenumbers, a wavenumber loop was implemented to run LIDORT or VLIDORT at a chosen resolution. Simulations at fine resolution require more time.

To have more information of the method used by LIDORT to calculate the radiative transfer budget and the radiances, please consult its User Manual (*lidort_3p3_userguide_v3_01nov07.pdf*).

6.1 Choice between LIDORT and VLIDORT

Both codes have their own advantages and disadvantages. The choice between them should mostly depend on the simulated spectrum. VLIDORT is capable of accounting for the polarization of radiation in the atmosphere. LIDORT, in its turn, is not able to account for polarization, but it consumes less operational memory than VLIDORT. The effects of polarization are known to be quite significant in UV, visible and near IR; while thermal emissions become important for wavelengths in excess of 2.4 μm (for Earth). Thus, it is advisable to use VLIDORT for the spectrum range before 2.4 μm (for Earth) and LIDORT in all other cases.

6.2 Aerosols

LIDORT and VLIDORT can account for scattering and absorption events caused by the presence of aerosols in an atmosphere. However, both codes need to be supplied with already calculated specific aerosol parameters, such as aerosol optical depth, single scattering albedo (SSA) and expansion (in scalar mode, Legendre) coefficients. Such an approach is not always convenient for the user, who faces the necessity to find a way to calculate these parameters outside the codes. Moreover, the majority of publications describe aerosols in the form of particle types and distributions, and not in the form of expansion coefficients.

To avoid this kind of possible inconvenience, it was decided to link both LIDORT and VLIDORT with aerosol processing codes. The list of best aerosol processing software available on the scientific market includes the SPHER and T-MATRIX codes, which are free to download from: <http://www.giss.nasa.gov/staff/mmishchenko/>.

These codes were developed by Dr. M. Mishchenko, NASA Goddard Institute for Space Studies, New York, USA. SPHER calculates scattering and absorption properties of polydisperse homogeneous spherical particles on the basis of the Lorenz-Mie theory [Mishchenko *et al.*, 2002]; while T-MATRIX is designed to be used for polydisperse, randomly oriented particles of identical axially symmetric shapes [Mishchenko *et al.*, 2002, Mishchenko & Travis, 1998].

Both SPHER and T-MATRIX handle several particle size distributions. The analytical formulas defining these distributions are presented below. In these formulas, $n(r)$ is the distribution function normalized to unity as follows:

$$\int_{r_1}^{r_2} n(r) dr \quad (6.1)$$

Here $n(r)dr$ is the fraction of particles with equivalent-sphere radii between r and $r + dr$, and $r1$ and $r2$ are the minimum and the maximum equivalent-sphere radii in the size distribution [Mishchenko & Travis, 1998].

6.2.1 Modified gamma distribution (both codes):

$$n(r) = \frac{1}{r_c \Gamma(\frac{\alpha+1}{\gamma})} \left(\frac{\alpha}{\gamma}\right)^{(\alpha+1)/\gamma} \left(\frac{r}{r_c}\right)^\alpha \exp\left(-\frac{\alpha}{\gamma} \left(\frac{r}{r_c}\right)^\gamma\right) \quad (6.2)$$

where Γ is the gamma function, α and γ are constants, and r_c is the radius value approximately corresponding to the peak of the distribution.

6.2.2 Log normal distribution (both codes)

$$n(r) = \frac{1}{(2\pi)^{1/2} r \ln \sigma_g} \exp\left(-\frac{(\ln r - \ln r_g)^2}{2(\ln \sigma_g)^2}\right) \quad (6.3)$$

where r_g and σ_g are the mean and standard deviations of r .

6.2.3 Power law distribution (both codes)

$$n(r) = \begin{cases} \frac{2r_1 r_2}{r_2^2 - r_1^2} r^{-3}, & \text{for } r_1 \leq r \leq r_2 \\ 0, & \text{otherwise} \end{cases} \quad (6.4)$$

For this distribution, the codes automatically calculate the minimum and the maximum radii r_1 and r_2 on the basis of supplied effective radius r_{eff} and effective variance v_{eff} which are analytically defined as:

$$r_{eff} = \frac{1}{G} \int_{r_1}^{r_2} n(r) r \pi r^2 dr \quad (6.5)$$

$$v_{eff} = \frac{1}{G r_{eff}^2} \int_{r_1}^{r_2} n(r) (r - r_{eff}) \pi r^2 dr \quad (6.6)$$

where

$$G = \int_{r_1}^{r_2} n(r) \pi r^2 dr \quad (6.7)$$

6.2.4 Gamma distribution (both codes)

$$n(r) = \frac{1}{ab\Gamma(\frac{1-2b}{\gamma b})} \left(\frac{r}{ab}\right)^{(1-3b)/b} \exp\left(-\frac{r}{ab}\right) \quad (6.8)$$

where Γ is the gamma function, and a and b are constants.

6.2.5 Modified power law distribution (both codes)

$$n(r) = \begin{cases} \frac{2r_1 r_2}{r_2^2 - r_1^2}, & 0 \leq r < r_1 \\ \frac{2r_1 r_2}{r_2^2 - r_1^2} \left(\frac{r}{r_1}\right)^\alpha, & \text{for } r_1 \leq r \leq r_2 \\ 0, & r > r_2 \end{cases} \quad (6.9)$$

6.2.6 Bimodal volume log normal distribution (SPHER only)

This distribution will be implemented in the near future.

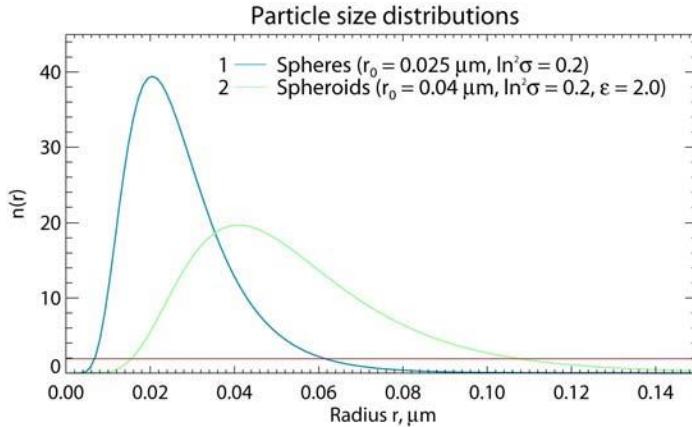


Figure 6: Example size distributions of Martian dust particles. Size distribution parameters were taken from [Dlugach et al., 2002].

Both SPHER and T-MATRIX have been slightly modified to become part of ALVL. In particular, their input parameters now come from outside the codes instead of being specified inside as before. Also, a special function was added to calculate the boundaries of a radius integration range on the basis of two accuracy parameters specified by the user. For example, let's consider the size distributions shown in Figure 6. There is an option to cut the “tails” of the distribution as shown with the red line. The minimum and the maximum radii r_1 and r_2 will be then calculated as values at the cross-points of the cut-line and the size distribution curves. The integration will be performed over a smaller range, which will result in a significant decrease in calculation time. The two accuracy parameters which the user has to specify are the value of cutoff for a given $n(r)$ and the accuracy of r_1 and r_2 calculation.

Aerosol parameters (extinction cross sections, SSA and expansion coefficients) do not change with wavenumber as rapidly as gas absorption parameters and, therefore, there is no need to calculate the aerosol parameters at the same fine wavenumber resolution as the gas parameters. There is an option in ALVL allowing the user to calculate the aerosol parameters at a coarse resolution and then to interpolate them to a finer resolution specified for absorption and RT calculations.

6.3 Surface

LIDORT/VLIDORT can simulate Lambertian and BRDF (Bidirectional Reflectance Distribution Function) surfaces. In case of a Lambertian surface the user has to input the surface albedo. In case of a BRDF surface the user is asked to choose up to 3 different BRDF kernels from those incorporated in the codes. The BRDF function will be then calculated as a linear combination of chosen semi-empirical kernel functions.

The list of BRDF kernels implemented in LIDORT and VLIDORT is shown in Table 6. The index is an internal code parameter which, nevertheless, has to be specified by the user. Unfortunately, the list of references for these kernels is not complete yet, as some kernels are stated “implemented” inside the codes, but the corresponding user’s guides do not have any record of them. The list will be updated after further research.

Name	Index (LIDORT)	Index (VLIDORT)	Reference
Lambertian	1	1	
Ross-thin	2	2	<i>Wanner et al., 1995</i>
Ross-thick	3	3	<i>Wanner et al., 1995</i>
Li-sparse	4	4	<i>Wanner et al., 1995</i>
Li-dense	5	5	<i>Wanner et al., 1995</i>
Hapke	6	6	<i>Hapke, 1993</i>
Roujean	7	7	<i>Wanner et al., 1995</i>
Rahman	8	8	<i>Rahman et al., 1993</i>
Cox-Munk	9	9	<i>Cox & Munk, 1954</i>
Gisssoil	10	13	
Gissnow	11	14	
GissCoxMunk	-	10	
Rherman	-	11	
Breon	-	12	

Table 6: List of BRDF kernels implemented in LIDORT/VLIDORT

In case of thermal spectrum, there is an option to calculate the surface albedo automatically on the basis of emission values. If the lower boundary of a given spectrum range is equal to/exceeds $2.4 \mu\text{m}$, the surface is automatically defined as Lambertian and its albedo is calculated as $1-\epsilon(v)$ (following the approach suggested by [Wan, 1999]).

Unfortunately, BRDF surface tests have not been successful for VLIDORT 2.4RT (latest version). For some reason, VLIDORT acts like surface albedo is 0, which is not correct. This issue is corrected in VLIDORT v. 3.5 issued in March 2011, F77 & F90, which will be soon incorporated in ALVL.

6.4 Comment on the Radiance units used in ASIMUT and (V)LIDORT

In ASIMUT all radiances are expressed in $\text{W.cm}^{-2}.\text{(cm}^{-1}\text{)}^{-1}.\text{sr}^{-1}$, whereas in LIDORT they are expressed in W.m^{-2} , thus integrated on a small wavenumber interval. The conversion factor is $\text{Radiance}_{\text{ASIMUT}} = \text{Radiance}_{\text{LIDORT}} * 10^4 * \text{Step}[\text{cm}^{-1}]$.

Moreover Lidort is working on Irradiances (factor of PI). All outputs from the (V)LIDORT part have thus been converted into radiances in $\text{W.cm}^{-2}.\text{(cm}^{-1}\text{)}^{-1}.\text{sr}^{-1}$.

6.5 (V)LIDORT input parameters

When running (V)LIDORT additional parameters have to be provided by the user. Some are grouped in a specific file using the format imposed by (V)LIDORT (file_LIDORT_flags). They are fully described in the *ASIMUT_Input_File.pdf* file. Some parameters must be given in the file.INP file and those used by SPHER/TMATRIX and defining the aerosols characteristics or optical properties are specified in an additional file also described in *ASIMUT_Input_File.pdf*.

6.5.1 Parameters defined in the file.INP

[SPn_FENn] fileLIDORTflags: Name of a file containing flags used to control RT.

[SPn_FENn] fileLIDORTlog: Name of output log file generated by LIDORT/VLIDORT only in case of error/warning.

[SPn_FENn] RTstreams: Number of quadrature streams in the cosine half space [0,1]; it must be less than or equal to the symbolic dimension MAXSTREAMS. Suggested values:

- 1 (no aerosol)
- 8 (with aerosol in IR)

[SPn_FENn] Nstokes: Number of Stokes vector components: choice between scalar and vector modes.

- 1 for LIDORT,
- 3 for VLIDORT

[SPn_FENn] transOnly: Flag for ignoring scattering and accounting for transmission only.

[SPn_FENn] rayleighOnly: Flag for simulating only Rayleigh scattering.

[SPn_FENn] SetSolarTo1: If SetSolarTo1=1, then the Solar spectrum is set to 1.0 (only for testing purposes).

7. Conclusions

The ASIMUT code performs forward modelling as well as retrieval of vertical information. Initially developed for nadir looking instruments such as IASI on board METOP-A and for Earth observations, ASIMUT has been extended to different geometries: off-axis nadir looking mode, solar occultations from space and from the ground. The latter option will be used for validation purposes using ground-based Fourier transform infrared (FTIR) instruments involved in the Network for the Detection of Atmospheric Composition Change (NDACC; <http://www.ndacc.org>). The internal structure of ASIMUT has been made such that it is easily adapted to simulate planetary atmospheres other than Earth (Mars and Venus).

The forward model includes the possibility to simulate LBL molecular absorption, but also the Rayleigh scattering, the absorption in the form of cross sections and continua of absorption due to H_2O , CO_2 , N_2 , and O_2 . The latter are essentially treated with the MT_CKD_1.2 formalism [20], with slight modifications to take into account new laboratory measurements. LBL simulation uses different line profile functions (Lorentz, Voigt, Galatry and Rautian). It is moreover possible to introduce a symmetric or asymmetric factor defining sub- or super-Lorentzian profiles. Spectroscopic parameters and absorption cross sections can be directly read from HITRAN, CDSD and GEISA databases, or from user defined files. The retrieval unit of ASIMUT uses the Optimal Estimation Method and is based on the analytical calculation of the Jacobians during the forward model run.

ASIMUT is continuously being updated and improved. In the near future, the treatment of solar lines present in direct sun measurements and the effect of direct sun reflection on the surface, for example, will be implemented. Based on the vertical optical depths obtained with ASIMUT, this algorithm uses an OEM to retrieve both the aerosol optical thickness and the surface temperature, assuming a thin horizontal layer of a mixture of two aerosol types located at an arbitrary altitude in the troposphere.

8. References

1. Cayla, F. and P. Javelle, *IASI instrument overview*. Int. Soc. Opt. Eng., Proc. SPIE, 1995. **2583**: p. 271– 281.
2. Senten, C., et al., *Technical Note: New ground-based FTIR measurements at Ile de La Réunion: observations, error analysis, and comparisons with independent data*. Atmospheric Chemistry and Physics, 2008(8): p. 3483– 3508.
3. De Mazière, M., et al. *Validation of IASI Atmospheric Chemistry Products for CO, O₃, HNO₃, N₂O and CH₄ with FTIR Ground-based network data*. in *1st EPS/MetOp RAO Workshop*. 2006. ESRIN, Frascati, Italy: ESA Publications SP-618.
4. Svedhem, H., et al., *Venus Express: The first European mission to Venus*. Planet. Space Sci., 2007. **55**(12): p. 1636-1652.
5. Mahieux, A., et al., *In-flight performance and calibration of SPICAV/SOIR on-board Venus Express*. Applied Optics, 2008. **47**(13): p. 2252-65.
6. Nevejans, D., et al., *Compact high-resolution space-borne echelle grating spectrometer with AOTF based on order sorting for the infrared domain from 2.2 to 4.3 micrometer*. Applied Optics, 2006. **45**(21): p. 5191-5206.
7. Rodgers, C., *Inverse methods for atmospheric sounding: Theory and practice*. World Scientific, ed. N.J. Hackensack. 2000: University of Oxford.
8. Vandaele, A.C., M. Kruglanski, and M.D. Mazière. *Modeling and retrieval of Atmospheric spectra using ASIMUT* in *Proc. of the First ‘Atmospheric Science Conference’, ESRIN, Frascati, Italy, 8 – 12 May 2006 (ESA SP-628, July 2006)*. 2006.
9. Spurr, R., T. Kurosu, and K. Chance, *A linearized discrete ordinate radiative transfer model for atmospheric remote-sensing retrieval*. J. Quant. Spectrosc. Radiat. Transfer, 2001. **68**(6): p. 689-735.
10. Spurr, R.J.D., *VLIDORT: A linearized pseudo-spherical vector discrete ordinate radiative transfer code for forward model and retrieval studies in multilayer multiple scattering media*. J. Quant. Spectrosc. Radiat. Transfer, 2006. **102**: p. 316-342.
11. Mishchenko, M.I. and L.D. Travis, *Capabilities and limitations of a current Fortran implementation of the T-matrix method for randomly oriented, rotationally symmetric scatterers*. J. Quant. Spectrosc. Radiat. Transfer, 1998. **60**(3): p. 309-324.
12. Bertaux, J.L., et al., *A warm layer in Venus' cryosphere and high altitude measurements of HF, HCl, H₂O and HDO*. Nature, 2007. **450**(29 November): p. 646-649, doi:10.1038/nature05974.
13. Kruglanski, M., et al. *Retrieval of formaldehyde from FTIR spectra: preliminary results*. in *NDACC IRWG meeting*. 2007. Puerto de La Cruz (Tenerife).
14. Vandaele, A.C., et al., *Composition of the Venus mesosphere measured by SOIR on board Venus Express*. J. Geophys. Res., 2008: p. doi:10.1029/2008JE003140.
15. Wilquet, V., et al., *Line parameters for the 01111-00001 band of ¹²C¹⁶O¹⁸O from SOIR measurements of the Venus atmosphere*. J. Quant. Spectrosc. Radiat. Transfer, 2008. **109**: p. 895-905.
16. Kurucz, R.L., et al., *Solar flux atlas from 296 nm to 1300 nm*, in *Natl. Sol. Obs. Atlas 1*. 1984, Harvard Univ., Cambridge, Mass.

17. Thuillier, G., et al., *The solar spectral irradiance from 200 to 2400 nm as measured by the SOLSPEC spectrometer from the ATLAS and EURECA missions*. Solar Physics, 2003. **214**: p. 1-22.
18. Gallery, W.O., F.X. Kneizys, and S.A. Clough, *Air mass computer program for atmospheric transmittance/radiance calculation: FSCATM*, H.A. Air Force Geophysical Laboratory, MA, Editor. 1983, Air Force.
19. Clough, S.A., et al., *Atmospheric spectral transmittance and radiance: FASCOD1B*. Proc. Soc. Photo Opt Instrum Eng, 1981. **277**: p. 152-166.
20. Clough, S.A., et al., *Atmospheric radiative transfer modeling: a summary of the AER codes*. J. Quant. Spectrosc. Radiat. Transfer, 2005. **91**: p. 233-244.
21. Meier, A., et al., *Improvements to air mass calculations for ground-based infrared measurements*. J. Quant. Spectrosc. Radiat. Transfer, 2004. **83**(1): p. 109-113.
22. Edlén, B., *The refractive index of air*. Metrologia, 1966. **2**(2): p. 71-80.
23. Ciddor, P.E., *Refractive index of air: new equations for the visible and near infrared*. Applied Optics, 1996. **35**(9): p. 1566-1573.
24. Tomasi, C., et al., *Improved algorithm for calculations of Rayleigh-scattering optical depth in standard atmospheres*. Applied Optics, 2005. **44**(16): p. 3320-3341.
25. Sneep, M. and W. Ubachs, *Direct measurement of the Rayleigh scattering cross section in various gases*. J. Quant. Spectrosc. Radiat. Transfer, 2005. **92**(3): p. 293-310.
26. *U.S. Standard Atmosphere*, 1976, U.S.G.P. Office, Editor. 1976: Washington, D.C.
27. Remedios, J.J., et al., *MIPAS reference atmospheres and comparisons to V4.61/V4.62 MIPAS level 2 geophysical data sets*. Atmos. Chem. Phys. Discuss., 2007. **7**(4): p. 9973-10017.
28. Goody, R.M., *Atmospheric Radiation*. 1964: Oxford University Press.
29. Fischer, J., et al., *Total internal partition sums for molecular species in the 2000 edition of the HITRAN database*. J. Quant. Spectrosc. Radiat. Transfer, 2003. **82**(1-4): p. 401-412.
30. Gamache, R.R., R.L. Hawkins, and L.S. Rothman, *Total internal partition sums for atmospheric molecules in the temperature range 70-2005 K: Atmospheric linear molecules*. J. Mol. Spectrosc., 1990. **142**(2): p. 205-219.
31. Kuntz, M., *A new implementation of the Humlincek algorithm for the calculation of the Voigt profile function*. J. Quant. Spectrosc. Radiat. Transfer, 1997. **57**(6): p. 819-824.
32. Wells, R.J., *Rapid approximation to the Voigt/Faddeeva function and its derivatives*. J. Quant. Spectrosc. Radiat. Transfer, 1999. **62**(1): p. 29-48.
33. Varghese, P.L. and R.K. Hanson, *Collisional narrowing effects on spectral line shapes measured at high resolution*. Applied Optics, 1984. **23**(14): p. 2376-2385.
34. Rautian, S.G. and I.I. Sobel'man, *The effect of collisions on the Doppler broadening of spectral lines*. Sov. Phys. Usp., 1967. **9**: p. 701-716.
35. Clough, S.A., F.X. Kneizys, and R. Davies, *Line shape and the water vapor continuum*. Atmos. Research, 1989. **23**(3-4): p. 229-241.
36. Pollack, J.B., et al., *Near-infrared Light from Venus' nightside: A spectroscopic analysis*. Icarus, 1993. **103**: p. 1-42.

37. Fomin, B.A., *Effective interpolation technique for line-by-line calculations of radiation absorption in gases*. J. Quant. Spectrosc. Radiat. Transfer, 1995. **53**(6): p. 663-669.

38. Rothman, L.S., et al., *The HITRAN 2004 Molecular Spectroscopic Database*. J. Quant. Spectrosc. Radiat. Transfer, 2005. **96**(2): p. 139-204.

39. Jaquinet-Husson, N., et al., *The 1997 spectroscopic GEISA databank*. JQSRT, 1999. **62**(2): p. 205-254.

40. Thibault, F., et al., *Infrared collision-induced absorption by O₂ near 6.4 μm for atmospheric applications: measurements and empirical modeling*. Applied Optics, 1997. **36**(3): p. 563-567.

41. Hermans, C., et al. *Absorption cross-section of the collision-induced bands of oxygen from the UV to the NIR*. in *NATO Advanced Research Workshop, Weakly Interacting Molecular Pairs: Unconventional Absorbers of Radiation in the Atmosphere*. 2002.

42. Maté, B., et al., *Absolute intensities for the O₂ 1.27 μm continuum absorption*. J. Geophys. Res., 1999. **104**(D23): p. 30585-30590.

43. Mlawer, E.J., et al., *Observed atmospheric collision-induced absorption in near-infrared oxygen bands*. J. Geophys. Res., 1998. **103**(D4): p. 3859-3863.

44. Greenblatt, G.D., et al., *Absorption measurements of oxygen between 330 and 1140 nm*. J. Geophys. Res., 1990. **95**(D11): p. 18577-18582.

45. Bernath, P., et al., *The Wulf bands of oxygen*. Chem. Phys. Letter, 1998. **297**(3-4): p. 293-299.

46. Fally, S., et al., *Fourier transform spectroscopy in the O₂ Herzberg bands: III. Absorption cross-sections of the collision-induced bands and of the Herzberg continuum*. J. Mol. Spectrosc., 2000. **204**(1): p. 10-20.

47. Yoshino, K., et al., *Improved absorption cross-sections of oxygen in the wavelength region 205-240 nm of the Herzberg continuum*. Planet. Space Sci., 1988. **36**(12): p. 1469-1475.

48. Borysow, A. and L. Frommhold, *Collision-Induced rototranslational absorption spectra of N₂-N₂ pairs for temperatures from 50 to 300 K*. The Astrophysical Journal, 1986. **311**: p. 1043-1057.

49. Boissoles, J., et al., *Theoretical calculation of the translation-rotation collision-induced absorption in N₂-N₂, O₂-O₂, and N₂-O₂ pairs*. J. Quant. Spectrosc. Radiat. Transfer, 2003. **82**(1-4): p. 505-516.

50. Lafferty, W., et al., *Infrared collision-induced absorption by N₂ near 4.3 μm for atmospheric applications: measurements and empirical modeling*. Applied Optics, 1996. **35**(30): p. 5911-5917.

51. Menoux, V., et al., *Collision-induced absorption in the fundamental band of N₂: temperature dependence of the absorption for N₂-N₂ and N₂-O₂ pairs*. Applied Optics, 1993. **32**(3): p. 263-268.

52. Mlawer, E.J., et al. *Collision-induced effects and the water vapor continuum*. in *Proc. of the 8th Atmospheric Radiation Measurement Science Team Meeting*. 1998. Tucson, Arizona, USA.

53. Magalhaes, J.A., J.T. Schofield, and A. Seiff, *Results of the Mars Pathfinder atmospheric structure investigation*. J. Geophys. Res., 1999. **104**(E4): p. 8943-8955.

54. Tashkun, S., et al., *CDSD-1000, the high-temperature carbon dioxide spectroscopic databank*. J. Quant. Spectrosc. Radiat. Transfer, 2003. **82**: p. 165-196.

55. Kobayashi, H., et al., *Development and evaluation of the interferometric monitor for greenhouse gases: a high-throughput Fourier-transform infrared radiometer for nadir Earth observation*. Applied Optics, 1999. **38**: p. 6801–6807.
56. Bernath, P.F., et al., *Atmospheric Chemistry Experiment (ACE): Mission overview*. Geophys. Res. Lett., 2005. **32**: p. doi:10.1029/2005GL022386.
57. Kauppinen, J. and P. Saarinen, *Line-shape distortions in misaligned cube corner interferometers*. Applied Optics, 1992. **31**(1): p. 69-74.
58. Hase, F., T. Blumenstock, and C. Paton-Walsh, *Analysis of the instrumental line shape of high-resolution Fourier transform IR spectrometers with gas cell measurements and new retrieval software*. Applied Optics, 1999. **38**(15): p. 3417-3422.
59. Goldman, A. and R.S. Saunders, *Analysis of Atmospheric Infrared Spectra for Altitude Distribution of Atmospheric Trace Constituents - I Method of Analysis*. J. Quant. Spectrosc. Radiat. Transfer, 1979. **21**(2): p. 155-161.
60. Kruglanski, M., A.C. Vandaele, and M. De Mazière. *Detection of tropospheric aerosols from satellite nadir sounding in the thermal infrared*. in EAC. 2005. Ghent, Belgium.

Annex 1: List of molecules supported and isotopic abundances

Isotopic Abundances Used for HITRAN

[based on P. De Bievre, N.E. Holden, and I.L. Barnes, "Isotopic Abundances and Atomic Weights of the Elements," *J.Phys.Chem.Ref.Data* **13**, 809-891 (1984)]

Molecule	Isot. Nb	Isot. code	Abundance
H₂O (1)	1	161	0.997317
	2	181	0.00199983
	3	171	0.000371884
	4	162	0.000310693
	5	182	0.000000623003
	6	172	0.000000115853
CO₂ (2)	1	626	0.984204
	2	636	0.0110574
	3	628	0.00394707
	4	627	0.000733989
	5	638	0.0000443446
	6	637	0.00000824623
	7	828	0.00000395734
	8	827	0.00000147180
	9	838	0.00000004446
	10	837	0.0000000165354
O₃ (3)	1	666	0.992901
	2	668	0.00398194
	3	686	0.00199097
	4	667	0.000740475
	5	676	0.000370237
N₂O (4)	1	446	0.990333
	2	456	0.00364093
	3	546	0.00364093
	4	448	0.00198582
	5	447	0.000369280
CO (5)	1	26	0.986544
	2	36	0.0110836
	3	28	0.00197822
	4	27	0.000367867
	5	38	0.000022225
	6	37	0.00000413292
CH₄ (6)	1	211	0.988274
	2	311	0.0111031
	3	212	0.000615751
	4	312	0.00000691785
O₂ (7)	1	66	0.995262
	2	68	0.00399141
	3	67	0.000742235
NO (8)	1	46	0.993974
	2	56	0.00365431

	3	48	0.00199312
SO₂ (9)	1	626	0.94568
	2	646	0.0419503
NO₂ (10)	1	646	0.991616
NH₃ (11)	1	4111	0.9958715
	2	5111	0.00366129
HNO₃ (12)	1	146	0.989110
OH (13)	1	61	0.997473
	2	81	0.00200014
	3	62	0.000155371
HF (14)	1	19	0.99984425
HCl (15)	1	15	0.757587
	2	17	0.242257
HBr (16)	1	19	0.506781
	2	11	0.493063
HI (17)	1	17	0.99984425
ClO (18)	1	56	0.755908
	2	76	0.241720
OCS (19)	1	622	0.937395
	2	624	0.0415828
	3	632	0.0105315
	4	623	0.00739908
	5	822	0.00187967
H₂CO (20)	1	126	0.986237
	2	136	0.0110802
	3	128	0.00197761
HOCl (21)	1	165	0.755790
	2	167	0.241683
N₂ (22)	1	44	0.9926874
HCN (23)	1	124	0.985114
	2	134	0.011076
	3	125	0.00362174
CH₃Cl (24)	1	215	0.748937
	2	217	0.239491
H₂O₂ (25)	1	1661	0.994952
C₂H₂ (26)	1	1221	0.977599
	2	1231	0.0219663
C₂H₆ (27)	1	1221	0.97699
PH₃ (28)	1	1111	0.99953283
COF₂ (29)	1	269	0.98654
SF₆ (30)	1	29	0.95018
H₂S (31)	1	121	0.949884
	2	141	0.0421369
	3	131	0.00749766
HCOOH (32)	1	126	0.983898
HO₂ (33)	1	166	0.995107
O (34)	1	6	0.997628
ClONO₂ (35)	1	5646	0.74957

	2	7646	0.239694
NO⁺ (36)	1	46	0.993974
HOBr (37)	1	169	0.505579
	2	161	0.491894
C₂H₄ (38)	1	221	0.977294
	2	231	0.0219595
CH₃OH (39)	1	2161	0.98593
CH3Br (40)	1	219	0.500995
	2	211	0.487433
CH3CN (41)	1	2124	0.973866
CF4 (42)	1	29	0.988890

New in HITRAN 2008 edition

Molecules only present in GEISA

Molecule	Isot. Nb	Isot. code	Geisa id	Abundance
GeH4 (43)	1	411	26	1.00
C3H8 (44)	1	221	28	1.00
C2N2 (45)	1	224	29	1.00
C4H2 (46)	1	211	30	1.00
HC3N (47)	1	124	31	1.00
C3H4 (48)	1	341	40	1.00

Molecules present in CFGL

Molecule	Isot. Nb	Isot. code	Cfgl id	Abundance
HONO (49)	1	1	24	1.00
HO2NO2 (50)	1	1	25	1.00
N2O5 (51)	1	1	26	1.00
CH3F (52)	1	1	29	1.00
CCL2F2 (53)	1	1	32	1.00
CCL3F (54)	1	1	33	1.00
CH3CCL3 (55)	1	1	34	1.00
CCL4 (56)	1	1	35	1.00
COCLF (57)	1	1	37	1.00
CHF2CL (58)	1	1	42	1.00
COCL2 (597)	1	1	43	1.00
CH3I (60)	1	1	45	1.00
CHCL2F (61)	1	1	48	1.00
OCIO (62)	1	1	58	1.00
F134A (63)	1	1	59	1.00
F142B (64)	1	1	61	1.00
CFC113 (65)	1	1	62	1.00
F141B (66)	1	1	63	1.00
C2H6PL (67)	1	1	66	1.00

PAN (68)	1	1	67	1.00
----------	---	---	----	------

Molecule ID in CFGL: H2O (1), CO2 (2), O3 (3), N2O (4), CO (5), CH4 (6), O2 (7), NO (8), SO2 (9), NO2 (10), NH3 (11), HNO3 (12), OH (13), HF (14), HCL (15), HBR (16), HI (17), CLO (18), OCS (19), H2CO (20), HOCL (21), HO2 (22), H2O2 (23), HONO (24), HO2NO2 (24), N2O5 (26), CLONO2 (27), HCN (28), CH3F (29), CH3CL (30), CF4 (31), CCL2F2 (321), CCL3F (33), CH3CCL3 (34), CCL4 (35), COF2 (36), COCLF (37), C2H6 (38), C2H4 (39), C2H2 (40), N2 (41), CHF2CL (42), COCL2 (43), CH3BR (44), CH3I (45), HCOOH (46), H2S (47), CHCL2F (48), HDO (49), SF6 (50), H218O (51), H217O (52), CH3D (53), O3668 (54), O3686 (55), O3667 (56), O3676 (57), OCLO (58), F134A (59), C3H8 (60), F142B (61), CFC113 (62), F141B (63), CH3OH (64), CH3CN (65), C2H6PL (66), PAN (67)

Annex 2: HITRAN format

HITRAN Database Format 1986 - 2001

The following table illustrates the format used for the spectroscopic parameters of the HITRAN database between 1986 and 2001.

Format for HITRAN Parameters, 1986 though 2001

Parameter	Molecule number	Isotopologue number	Transition wavenumber (cm ⁻¹)	Line Intensity	R_p^2	Air-broadened width	Self-broadened width	lower-state Energy	Temperature dependence (of air width)	Pressure shift	upper vibrational quanta	lower vibrational quanta	upper local quanta	lower local quanta	Error codes	Reference codes
Field Length	2	1	12	10	10	5	5	10	4	8	3	3	9	9	3	6

HITRAN Database Format 2004-present

The following table illustrates the format used for the spectroscopic parameters starting with the HITRAN2004 Edition. This format is 160 characters per record (line transition). Note the additional parameters compared to earlier editions.

Format for HITRAN Parameters, Editions after 2001

Parameter	Molecule number	Isotopologue number	Transition wavenumber (cm ⁻¹)	Line Intensity	Einstein A-coefficient	Air-broadened width	Self-broadened width	lower-state Energy	Temperature dependence (of air width)	Pressure shift	upper vibrational quanta	lower vibrational quanta	upper local quanta	lower local quanta	Error codes	Reference codes	Flag for line-mixing	upper statistical weight	lower statistical weight
Field Length	2	1	12	10	10	5	5	10	4	8	15	15	15	15	6	12	1	7	7
Data type	Integer	Integer	Real	Real	Real	Real	Real	Real	Real	Real	Text	Text	Text	Text	Integer	Integer	Text	Real	Real

Format for the IR cross sections:

Infrared Molecular Absorption Cross-sections

In the folder IR_XSECT\ are placed files of IR cross-sections, the definition and units being described in previous articles on the database.^a Each portion of the cross-section files, i.e., a set with a temperature and pressure pair, contains a header that points to the reference for that observation and other relevant information. The sets contain absorption cross-sections that are in equal wavenumber (cm⁻¹) increments, and the intervals can be determined by the minimum and maximum wavenumber given in the header and the number of points, i.e.,

$$\Delta V = (V_{max} - V_{min})/(npts - 1),$$

where V_{max} is the maximum wavenumber of the set, V_{min} is the minimum wavenumber of the set, and $npts$ is the number of points in the set. The new format is quite similar to what we had before, except that the field length for the chemical symbol has been increased from 10 to 20 (there is already one compound that requires 13 characters), the number of points and temperature fields have been reduced from 10 to 7, and the pressure field has been reduced to 6. Previously the last 30 characters were rather undefined, although there was an indicator for the reference. Now we have put in the resolution of the observation. For Fourier transform spectrometer observations (all of the IR data), this will be in wavenumber. For grating spectrometer observations (presently only for some UV data), we employ xxxmA (where xxx is a number and mA indicates milli-Angstroms). Since some of the experiments were done in air, we have a field in which we put "air" (blank indicates "pure gas"). Finally, there is an integer pointer at the end for the references, which we will manage in the HAWKS software as well as providing an acrobat file (/Documentation/ReferenceTable.pdf) for those not using HAWKS.

Following is the format of the headers in the current edition:

"2000" Cross-section Header format:

Chemical symbol	Wavenumber	Number	Temp.	Press.	Maximum	Res.	Common Name	Not	Bro.	Ref.
	Minimum	Maximum	Of pts.	[K]	[Torr]	X-section		used	ad.	No
20	10	10	7	7	6	10	5	15	4	3
	10	20	30	40	50	60	70	80	90	100

Note: Chemical Symbol is right adjusted; Res. is resolution in cm⁻¹ for FTS measurements, and in milli-Angstroms for grating measurements in the UV (xxmA)

Annex 3: GEISA format

Format description for the GEISA 2003 database:

Parameter	A	B	C	D	E1	E2	E3	E4	F	G	I	J	K	L	M	N	O	P	Q
Field length	12	11	6	10	9	9	9	9	4	3	3	3	2	1	10	5	8	3	6
Fortran descriptor	F12.6	1PD11.4	0PF6.4	F10.4	A9	A9	A9	A9	F4.2	I3	I3	A3	I2	I1	1PE10.3	0PF5.4	F8.6	I3	I6

R	A'	B'	C'	F'	O'	R'	N'	S	S'	T	T'	U	U'
6	10	11	6	4	8	6	5	4	4	8	8	4	4
F6.4	F10.6	1PD11.4	0PF6.4	F4.2	F8.6	F6.4	F5.4	F4.2	F4.2	F8.6	F8.6	F4.2	F4.2

(F12.6,D11.4,F6.4,F10.4,A36,F4.2,I3,I3,A3,I2,I1,E10.3,F5.4,F8.6,I3,I6,F6.4,F10.6,D11.4,F6.4,F4.2,F8.6,F6.4,F5.4,F4.2,F4.2,F8.6,F8.6,F4.2,F4.2)

A B C D E F G I J K L M N O P Q R A' B' C' F' O' R' N' S S' T
T' U U'

A: wavenumber (cm-1) of the line

B: intensity of the line in (cm-1/(molecule.cm-2) @296K

C: Air broadening pressure halfwidth (HWHM) (cm-1atm-1) @296K

D: Energy of the lower transition level (cm-1)

E: Transition quantum identifications for the lower and upper state of the transition

F: temperature dependence coefficient n of the air broadening halfwidth

G: identification code for isotope as in GEISA

I: identification code for molecule as in GEISA

J: Internal GEISA code for the data identification

K: Molecule number as in HITRAN

L: isotope number (1=most abundant, 2= second...etc) as in HITRAN

M: transition probability (in debye2)

N: self broadening pressure halfwidth (HWHM) (cm-1atm-1) @296K (for water)

O: air pressure shift of the line transition (cm-1atm-1) @296K

P: accuracy indices for frequency, intensity and halfwidth

Q: indices for lookup of references for frequency, intensity and halfwidth

R: temperature dependence coefficient n of the air pressure shift

A': estimated accuracy (cm-1) on the line position

B': estimated accuracy on the intensity of the line in (cm-1/(molecule.cm-2)

C': estimated accuracy on the air collision halfwidth (HWHM) (cm-1atm-1)

F': estimated accuracy on the temperature dependence coefficient n of the air broadening halfwidth

O': estimated accuracy on the air pressure shift of the line transition (cm-1atm-1) @296K

R': estimated accuracy on the temperature dependence coefficient n of the air pressure shift

N': estimated accuracy on the self broadened (HWHM) (cm-1atm-1) @296K (for water)

S: temperature dependence coefficient n of the self broadening halfwidth (for water)

S': estimated accuracy on the temperature dependence coefficient n of the self broadening halfwidth (for water)

(for water)

T: self pressure shift of the line transition (cm-1atm-1) @296K (for water)

T': estimated accuracy on the self pressure shift of the line transition (cm-1atm-1) @296K (for water)

U: temperature dependence coefficient n of the self pressure shift (for water)

U': estimated accuracy on the temperature dependence coefficient n of the self pressure shift (for water)

And for the Cross Sections:

Absorption cross sections data format :

Parameter	Wavenumber (cm ⁻¹)	Absorption cross-section (cm ⁻² mol ⁻¹)	Molecule identification code	<u>Atmospheric conditions code</u>
Field length	12	10	3	3
Fortran descriptor	F12.6	1PE10.3	I3	I3

Associated file for the atmospheric conditions code format :

Parameter	<u>Atmospheric conditions code</u>	Temperature (K)	Presssure (Pa)	Resolution (cm ⁻¹)	Internal GEISA code for data identification
Field length	3	9	12	7	1
Fortran descriptor	I3	F9.1	F12.2	F7.3	A1

Annex 4: Radiance – irradiance

Radiance = energy emitted per unit area (m^2),
 per unit time (s)
 per unit wavelength (nm, or μm , or cm^{-1})
 per incremental solid angle (sr)

Unit of radiance = $\text{W} \cdot \text{m}^{-2} \cdot \mu\text{m}^{-1} \cdot \text{sr}^{-1}$

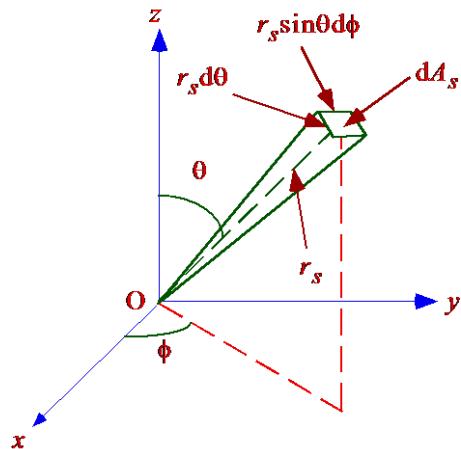
Ex = blackbody function, $B(\lambda, T)$

$$B_{\lambda,T} = \frac{2hc^2}{\lambda^5 \left[\exp\left(\frac{hc}{\lambda k_B T}\right) - 1 \right]}$$

Incremental solid angle : incremental surface area on a unit sphere

$$d\Omega = dA_S / r_S^2$$

with dA_S = an incremental surface area
 r_S = radius of the true sphere



I_λ = **spectral radiance** at a given point in space, regardless of the source of the radiance

Unit = $\text{W} \cdot \text{m}^{-2} \cdot \mu\text{m}^{-1} \cdot \text{sr}^{-1}$

E_λ = **spectral actinic flux** : integral of the spectral radiance over all solid angles of a sphere

$$dE_\lambda = I_\lambda \cdot d\Omega$$

if I isotropic : $E_\lambda = 4\pi \cdot I_\lambda$

Unit = $\text{W} \cdot \text{m}^{-2} \cdot \mu\text{m}^{-1}$.

F_λ = **spectral irradiance (or net flux, or energy flux)** : vertical component of the radiant energy propagating from all directions across a flat surface per unit area, time and wavelength. Integral of the spectral radiance over the hemisphere above the x-y plane

For $\text{sza} = \theta = 0^\circ$: irradiance impinging the x-y plane is maximal
 $\text{Sza} = \theta = 90^\circ$: irradiance is 0.0

$$dF_\lambda = I_\lambda \cdot \cos \theta \cdot d\Omega$$

if I isotropic : $F_\lambda = \pi \cdot I_\lambda$

⇒ **isotropic spectral irradiance = π x isotropic radiance**

Unit = $\text{W} \cdot \text{m}^{-2} \cdot \mu\text{m}^{-1}$.

Effect of the distance from the Sun- radiance at top of atmosphere

$F_{p,\lambda}$ = radiance emitted at the Sun's photosphere = $B(\lambda, T_p)$, with T_p the temperature of the Sun's photosphere (5796 K).

$F_{\text{Planet},\lambda}$ = incident solar radiation at the top of the planet's atmosphere

Depends on: The solar radiance

The planet-Sun distance (Radiance decreases proportionally to the inverse square distance from source)

$$F_{\text{Planet},\lambda} = (r/R)^2 \cdot F_{p,\lambda}$$

With r = radius of the Sun (at the photosphere) (6.96×10^8 m)

R = Planet-Sun distance

Units of radiance/irradiance

How to pass from $\text{W} \cdot \text{m}^{-2} \cdot \text{nm}^{-1} \cdot \text{sr}^{-1}$ to $\text{W} \cdot \text{cm}^{-2} \cdot (\text{cm}^{-1})^{-1} \cdot \text{sr}^{-1}$, used in ASIMUT

$$B_\lambda d\lambda = \nu^2 B_\nu d\nu$$

$$B_\nu = \frac{B_\lambda}{\nu^2} \quad \frac{W}{m^2 \cdot nm \cdot (cm^{-1})^2} = \frac{B_\lambda}{\nu^2} \quad \frac{W}{m^2 \cdot (10^{-9} m) \cdot (cm^{-1}) \cdot (cm^{-1})} = \frac{B_\lambda}{\nu^2} \quad \frac{W \cdot (10^{-2} m)}{(10^{-2} cm)^2 \cdot (10^{-9} m) \cdot (cm^{-1})}$$

$$B_\nu = \frac{B_\lambda}{\nu^2} 10^3 \quad \frac{W}{cm^2 \cdot (cm^{-1})}$$

How to pass from $\text{W} \cdot \text{cm}^{-2} \cdot \text{nm}^{-1} \cdot \text{sr}^{-1}$ to $\text{W} \cdot \text{cm}^{-2} \cdot (\text{cm}^{-1})^{-1} \cdot \text{sr}^{-1}$, used in ASIMUT

$$B_\lambda d\lambda = \nu^2 B_\nu d\nu$$

$$B_\nu = \frac{B_\lambda}{\nu^2} \frac{W}{cm^2 \cdot nm \cdot (cm^{-1})^2} = \frac{B_\lambda}{\nu^2} \frac{W}{cm^2 \cdot (10^{-9} m) \cdot (cm^{-1}) \cdot (cm^{-1})} = \frac{B_\lambda}{\nu^2} \frac{W \cdot (10^{-2} m)}{cm^2 \cdot (10^{-9} m) \cdot (cm^{-1})}$$

$$B_\nu = \frac{B_\lambda}{\nu^2} 10^7 \frac{W}{cm^2 \cdot (cm^{-1})}$$

In ASIMUT

1/ Reading the input file

Input file

[Solar]

FileSolar = Filename

This file must contain at least 2 columns :

ν [cm⁻¹], irradiance at the top of the Earth atmosphere [W/cm²/(cm⁻¹)]

!! irradiance (not radiance), at top of the Earth atmosphere:

$$\text{Irradiance}(\nu) = \pi \cdot (r/R)^2 \cdot B(\nu, T_s)$$

with T_s : Temperature of Sun photosphere

r : Sun radius

R : distance between Earth and Sun

$B()$: Blackbody (**Radiance**) [W/cm²/(cm⁻¹)/sr]

Example of files : Solar_irradiance_ACE.dat or Solar_irradiance_SOLSPEC.dat built using ACE_irradiance.m

2/ Building the I_{Sun} vector

```
int Sun ( double *Io, double *Wave, long NbWave,
          double resol, FENETRE *Fenetre, double Tsun )
```

Reads the irradiance from the file specified in the input file and divides it by EARTHSOLANGLE (or Planet)

which is $= \pi \cdot (r/R_{E-S})^2$ with R_{E-S} = distance Planet-Sun.

$$\begin{aligned} I_{Sun} &= \text{Irradiance_from_File} / \text{EARTHSOLANGLE} \\ &= \text{Irradiance_from_File} / (\pi \cdot (r/R_{E-S})^2) \\ &\sim B(\nu, T_s) \end{aligned}$$

3/ In the RT part

When taking into account the reflection of the Sun radiation on the planet's surface:

Cf for ex Haus and Titov {Haus, 2000 #2110}

Lower boundary condition equ 15 without diffusion

$$I^{BC}(z_{BC}, +\mu, \phi) = \frac{a}{\pi} E^\odot \mu_0 \tau(z_t, z_b, \mu_0) + \varepsilon B(T_{Surface})$$

with E^\odot = solar irradiance at the top of the atmosphere of the planet
 $= \pi \cdot (r/R_{P-S})^2 \cdot I_{Sun} = R_{Sun}$

$$\mu_0 = \cos(sza);$$

R_{P-S} = distance Planet-Sun;

a = albedo of the planet (if not specified : $a = 1 - \varepsilon$)

ε = emissivity of the planet

`Rsun = Isun[iWav] * Fenetre->AngleSolar;`

Where $Fenetre->AngleSolar = \pi \cdot (r/R_{P-S})^2$.

4/ Definition of the Transmittance

$$\text{Transmittance} = \frac{\text{Radiance}}{I_0}$$

In the normal case I_0 = radiance from the source

= radiance from the sun if looking at the sun (solar occultation), of the planet if looking at the planet in Nadir mode when no solar reflection is taken into account.

If solar reflection on the surface is included :

$$\text{Transmittance} = \frac{\text{Radiance} \times \pi}{E^\odot} = \text{RadianceFactor}$$

Annex 5: Physical constants

The physical constants used throughout ASIMUT are defined in the file *asimut.h*. They have been updated according to the updates given on the [NIST website](#). In June 2012, the constants used in ASIMUT were the following:

```

#define PI 3.1415926535897932384626433832795 /*!< PI */
#define PIX2 PI*2.0 /*!< = 2 PI */
#define PI2 9.869604401089358 /*!< = PI^2 */
#define PI3 31.0062766802998 /*!< = PI^3*/
#define PIDEMI 1.570796326794897 /*!< = PI/2*/

#define SQRTPI 1.772453850905516 /*!< = sqrt(PI)*/
#define ISQRTPI 0.5641895835477563 /*!< = 1/sqrt(PI)*/

#define SQRT2 1.414213562373095 /*!< = sqrt(2.0)*/
#define CONST1 8.325546111576977e-001 /*!< = sqrt(ln(2))*/
#define CONST2 CONST1*ISQRTPI /*!< = sqrt(ln(2)/pi)*/

#define CLUM 2.99792458e+08 /*!< ligth velocity (m/s)*/
#define HPL 6.62606957e-34 /*!< Planck cst (Js)*/
#define T0 273.15 /*!< STP Temperarture (K)*/
#define P0 1013.25 /*!< STP Pressure (hPa)*/
#define AVOGAD 6.02214129e+23 /*!< Avogadro Na (mol-1)*/
#define KB 1.3806488e-23 /*!< Boltzmann cst (J.mol-1)*/
#define RADCN2 1.4387752 /*!< =(2*KB) / (h*c)*/
#define LOSCH 2.686780515696967e+19 /*!< Loschmidt number =Na.P0 / R.T0 (molec/cm3)*/
#define GCAIR 1.380648792449704e-19 /*!< =Rgaz/Na =1.0/(LOSCH*T0/P0)*/

#define RGZ 8.3144621 /*!< Gas Constant = J mol-1K-1*/

#define PLANCKC1 2.0e4*HPL*CLUM*CLUM /*!< Planck cst C1 (W.cm2/sr) */
#define PLANCKC2 HPL*CLUM*1.0e2/KB /*!< Planck cst C2 (K/cm-1) = 1.438751926703556 */

#define DEG2RAD 0.01745329251994330 /*!< =(PI/180) */
#define RAD2DEG 57.2957795130823200 /*!< =(180/PI) */

/* Planet Radius calculation */
#define RE_EARTH 6371.23 /*!< Earth radius (km)*/
#define REM1 6378.39 /*!< Earth radius Model Atm 1 (km)*/
#define REM4 6356.91 /*!< Earth radius Model Atm 4,5 (km)*/
#define REQ 6378137 /*!< major axis (Lat,Lon corr) (m)*/
#define REP 6356752.3 /*!< minor axis (Lat,Lon corr) (m)*/
#define E2 6.694384442042606e-003/*!< =(REQ*REQ-REP*REP) / (REQ*REQ) */
#define EP2 6.739501254387102e-003/*!< =(REQ*REQ-REP*REP) / (REP*REP) */

#define RE_MARS 3389.9 /*!< Mars radius (km)*/
#define RE_VENUS 6051.8 /*!< Venus radius (km) */

#define TEMPEARTH 288.0 /*!< Temperature of Earth's surface (K)*/
#define TEMPmars 215.0 /*!< Temperature of Mars's surface (K)*/
#define TEMPVENUS 733.0 /*!< Temperature of Venus's surface (K)*/
#define TEMPSOL 5780.0 /*!< Temperature of Sun's surface (K) */

#define SUNRADIUS 696000.0 /*!< Radius of Sun (km)*/
#define AU 149599000.0 /*!< dist Earth-Sun = 1 au (km)*/
#define SUN_EARTH 1.0 /*!< distance Sun-Earth (au)*/
#define SUN_VENUS 0.72 /*!< distance Sun-Venus (au)*/
#define SUN_MARS 1.52 /*!< distance Sun-Mars (au) */

```

Annex 6: Template of file for a new instrument

Template of the NEWINSTRUMENT.c file

```

/*****
* 
* NEWINSTRUMENT.c
* DESCRIPTION : Functions relative to NEWINSTRUMENT
*
***** */

/* Read the header */

/*****
* Name of Author
* Created : date
*
* $Rev::: 337      $
* Last modified : $Date::: 2011-04-15 16:26:52#$
* by $Author::: xxxx $ 
***** */

#include "../common/Asimut.h"

/*****
* 
* FUNCTION    : Read_NEWINST    (char *FileName,
*                           INSTRUMENT *Instrument,
*                           GEOMETRY *Geo,
*                           int *ZPTfromSpectrum)
*
* PURPOSE     : Reading the NEWINSTRUMENT spectra from FileName
*
* INPUT       : FileName : name of NEWINSTRUMENT file
*               Instrument : Instrument structure
*               Geo : Geometry of the measurement
*               ZPTfromSpectrum : =1 if ZPT information in this file
*
* OUPUT       : Matrix containing the spectrum (wavenb,Intensity)
*
***** */

Matrix *Read_NEWINST (      char      *FileName,
                           INSTRUMENT *Instrument,
                           GEOMETRY  *Geo,
                           int       *ZPTfromSpectrum )
{
Matrix      *Data=NULL;
double      **Matrice=NULL;
long        nbPts, nbColumns=0;
char        str[1024], Line[256];
FILE        *hFile=NULL;
int         onError=ERR_NOERROR;

/* Read the header */
if ( (hFile=fopen(FileName,"rb")) == NULL )
{
sprintf(str,"NEWINSTRUMENT file %s does not exist !", FileName);
set_Error(ERR_FILE_NONE, SEV_STOPPROG, __LINE__, __FILE__,str);
return (NULL);
}
fgets ( Line, 255, hFile );
fgets ( Line, 255, hFile );

```

```

        fgets ( Line, 255, hFile );
...
        fgets ( Line, 255, hFile );

        fclose(hFile);
        hFile=NULL;

/* Read the data */
/* for example using the Read_Ascii function or the following one*/
if (Read_Ascii_from (FileName, &Matrice, &nbPts, &nbColumns,
    nbofLineinTheHeader) !=ERR_NOERROR )
    {onError=ERR_REPORTING; errout.Line=__LINE__;
    sprintf( errout.Message,"Problem reading NEWINSTRUMENT File %s !",
    FileName);
    goto TheEnd; }

if ( nbColumns<2 )
{onError=SEV_STOPPROG; errout.Line=__LINE__;
    sprintf( errout.Message,"Not enough columns in NEWINSTRUMENT File %s !"
    ,FileName);
    goto TheEnd; }

/* be sure that the wavenumber scale is ascending */
CorrectSpectra (Matrice, nbPts, nbColumns);

/* create the MATRIX structure */
if ( (Data = Merge_Matrix (Matrice[0], Matrice[1], nbPts)) == NULL )
    {onError=ERR_REPORTING; errout.Line=__LINE__;
    sprintf( errout.Message,"Problem Merge_Matrix");
    goto TheEnd; }

/* The step is also initialized here */
Instrument->Step = (Matrice[0][nbPts-1]-Matrice[0][0])/(nbPts-1);

/* this flag indicate if ZPT information (file name) is present or not */
*ZPTfromSpectrum = NON;

TheEnd:
    free_Matdbl2 (&Matrice,nbColumns);
    if (hFile!=NULL) fclose(hFile);
    if (onError==ERR_NOERROR)
        return Data;
    else
    {
        free_Matrix(&Data);
        set_Error(onError, errout.Severity,errout.Line, __FILE__,errout.Message);
        return(NULL);
    }
}

/* The following are not needed if the LIST option is not required */

*****
*   FUNCTION    : GetNbSpec_NEWINST_LIST ( char *FileName,
*                                         int *SamePass,
*                                         int *First)
*
*   PURPOSE     : Get number of spectra from FileName containing a list of
*                 spectra
*
*   INPUT       : FileName : name of list file

```

```

*           SamePass : = 1 if all spectra are analysed simultaneously  *
*           First : index to the first spectrum
*
*   OUPUT      : nb of spectra
*
***** */
int GetNbSpec_NEWINST_LIST ( char *FileName, int *SamePass, int *First )
{
int nb=0;
char Line[NL];
FILE *hFile;
char str[1024];

*First = 0;

if ( (hFile=fopen(FileName,"rb")) == NULL )
{
sprintf(str,"Can not open file %s !", FileName);
set_Error(ERR_FILE_NONE, SEV_STOPPROG, __LINE__, __FILE__,str);
return (0);
}

while (fgets ( Line, NL-1, hFile ) )
{
nb++;
}

(*SamePass) = NON; /* all spectra are NOT analysed simultaneously*/

fclose (hFile);
return (nb);
}

***** */
*   FUNCTION   : GetNameSpec_NEWINST_LIST ( char *FileName,
*                                         int index,
*                                         char **strN,
*                                         char **Desc )
*
*   PURPOSE    : Get file name and description of spectrum index
*
*   INPUT      : FileName : name of list file
*                 index: index of the spectrum in the list
*                 strN : Complete filename to spectrum with path
*                 Desc : description of the file in the list
*
*   OUPUT      : type of error
*
***** */
int GetNameSpec_NEWINST_LIST ( char *FileName, int index, char **strN, char
**Desc )
{
int nb, i;
char str[1024], Line[NL], *path=NULL, *base=NULL, *stnb=NULL;
FILE *hFile;
int onError=ERR_NOERROR;

if ( (hFile=fopen(FileName,"rb")) == NULL )
{
sprintf(str,"Can not open file %s !", FileName);
return (set_Error(ERR_FILE_NONE, SEV_STOPPROG, __LINE__, __FILE__,str));
}

```

```

    }

for (i=0;i<=index;i++)
    fgets ( Line, NL-1, hFile );

stnb=trim_ext(Line);
if ( allocini_String(Desc,stnb)!=ERR_NOERROR )
    {onError=SEV_STOPFCT; errout.Line=__LINE__;
     sprintf( errout.Message,"Problem Desc"); goto TheEnd;}

/* the name of the next spectrum is the first thing read */
*strN=Extract_1s(stnb);
free_String (&stnb);

if ( allocini_String(strN,str)!=ERR_NOERROR )
    {onError=SEV_STOPFCT; errout.Line=__LINE__;
     sprintf( errout.Message,"Problem strN"); goto TheEnd;}

TheEnd:
if ( hFile!=NULL) fclose (hFile);
free_String (&stnb);
if ( onError )
    set_Error(ERR_REPORTING,errout.Severity, errout.Line,
              __FILE__,errout.Message);
return (onError);
}

/*********************************************
*
*  FUNCTION    : ModifySpec_NEWINST_LIST  ( SPECTRUM      *Spectrum,
*  *              char          *ModDesc,
*  *              INSTRUMENT   *Inst,
*  *              GEOMETRY     *Geo,
*  *              int          *ZPTfromSpectrum )
*
*  PURPOSE     : Modify Spectrum, INstrument, or Geometry in accordance
*  *              to the string ModDesc
*
*  INPUT       : Spectrum : Spectrum structure
*  *              ModDesc : Descriptio of the measurement read from the list
*  *              file
*  *              Inst : Instrument structure
*  *              Geo : Geometry structure
*  *              ZPTfromSpectrum : =1 if ZPT information in this file
*
*  OUPUT       : type of error
*
********************************************/
int ModifySpec_NEWINST_LIST  ( SPECTRUM      *Spectrum,
                               char          *ModDesc,
                               INSTRUMENT   *Inst,
                               GEOMETRY     *Geo,
                               Int          *ZPTfromSpectrum )
{
char  *str;
int   nb;

if ( allocini_String(&str,ModDesc)!=ERR_NOERROR )
    return(set_Error(ERR_REPORTING,
                     errout.Severity,__LINE__,__FILE__,errout.Message));

/* index useful information from the ModDesc string*/

```

```

*ZPTfromSpectrum = NON;

free_String (&str);
return( ERR_NOERROR);
}

/* The following is not needed if the OCC option is not required */

/***************************************************************
*
*  FUNCTION    : GetNbSpec_NEWINST_OCC (char *FileName,
*                                         int *SamePass,
*                                         int *First)
*
*  PURPOSE     : Get number of spectra from FileName containing a list of
*                 spectra defining an occultation
*
*  INPUT       : FileName : name of list file
*                 SamePass : = 1 if all spectra are analysed simultaneously
*                 First : index to the first spectrum
*
*  OUPUT       : nb of spectra
*
***************************************************************/
int GetNbSpec_NEWINST_OCC ( char *FileName, int *SamePass, int *First )
{
int nb=0;
char Line[NL];
FILE *hFile;
char str[1024];

*First=0;

if ( (hFile=fopen(FileName, "rb")) == NULL )
{
sprintf(str,"Can not open file %s !", FileName);
set_Error(ERR_FILE_NONE, SEV_STOPPROG, __LINE__, __FILE__,str);
return (0);
}

while (fgets ( Line, NL-1, hFile ))
{
nb++;
}

(*SamePass) = YES; /* all spectra analysed simultaneously*/

fclose (hFile);
return (nb);
}

```

Template for the NEWINSTRUMENT.h file

```
*****  
*  
* NEWINSTRUMENT.h  
* DESCRIPTION : all functions relatives to NEWINSTRUMENT  
*  
* This program is part of the ASIMUT Project  
*  
*****  
  
/*! \file NEWINSTRUMENT.h  
 \brief Functions relative to NEWINSTRUMENT  
  
 Created : 07/03/2006  
 $Rev::: 257  
 Last modified : $Date::: 2009-10-01 16:31:12#$  
 by $Author::: xxx  
 */  
  
#ifndef __newinstrument_h  
#define __newinstrument_h  
  
/*! \fn Read_NEWINST ( char *FileName,  
 INSTRUMENT *Instrument,  
 GEOMETRY *Geo,  
 int *ZPTfromSpectrum )  
 \brief Reading the IMG spectra from FileName  
 \param[in] FileName : name of IMG file  
 \param[in] Instrument : Instrument structure  
 \param[in] Geo : Geometry of the measurement  
 \param[in] ZPTfromSpectrum : =1 if ZPT information in this file  
 \return Matrix containing the spectrum (wavenb,Intensity)  
 */  
Matrix *Read_NEWINST ( char *FileName,  
 INSTRUMENT *Instrument,  
 GEOMETRY *Geo,  
 int *ZPTfromSpectrum );  
  
/*! \fn GetNbSpec_NEWINST_LIST ( char *FileName, int *SamePass, int *First )  
 \brief Get number of spectra from FileName containing a list of  
 spectra  
 \param[in] FileName : name of list file  
 \param[out] SamePass : = 1 if all spectra are analysed simultaneously  
 \param[out] First : index to the first spectrum  
 \return nb of spectra  
 */  
int GetNbSpec_NEWINST_LIST ( char *FileName, int *SamePass, int *First );  
  
/*! \fn GetNameSpec_NEWINST_LIST( char *FileName, int index,  
 char **strN, char **Desc )  
 \brief Get file name and description of spectrum index  
 \param[in] FileName : name of list file  
 \param[in] index: index of the spectrum in the list  
 \param[in] strN : Complete filename to spectrum with path  
 \param[in] Desc : description of the file in the list  
 \return type of error  
 */  
int GetNameSpec_NEWINST_LIST ( char *FileName, int index, char **strN,  
 char **Desc );
```

```

/*! \fn ModifySpec_NEWINST_LIST ( SPECTRUM    *Spectrum,
                                char        *ModDesc,
                                INSTRUMENT  *Inst,
                                GEOMETRY   *Geo,
                                Int         *ZPTfromSpectrum )
\brief Modify Spectrum, Instrument, or Geometry in accordance
       to the string ModDesc
\param[in] Spectrum : Spectrum structure
\param[in] ModDesc : Descriptio of the measurement read from the list
                    file
\param[in] Inst : Instrument structure
\param[in] Geo : Geometry structure
\param[in] ZPTfromSpectrum : =1 if ZPT information in this file
\return type of error
*/
int ModifySpec_NEWINST_LIST ( SPECTRUM    *Spectrum,
                                char        *ModDesc,
                                INSTRUMENT  *Inst,
                                GEOMETRY   *Geo,
                                Int         *ZPTfromSpectrum );

/*! \fn GetNbSpec_NEWINST_OCC (char *FileName,int *SamePass,int *First)
\brief Get number of spectra from FileName containing a list of
       spectra defining an occultation
\param[in] FileName : name of list file
\param[in] SamePass : = 1 if all spectra are analysed simultaneously
\param[in] First : index to the first spectrum
\return nb of spectra
*/
int GetNbSpec_NEWINST_OCC ( char *FileName, int *SamePass, int *First );

#endif

```

Annex 7: Contacts and rights of use

Please contact the Belgian Institute for Space Aeronomy to obtain the access rights to use ASIMUT.

Contacts:

Ann C. Vandaele :	a-c.vandaele@aeronomie.be
	Tel: +32-2-373-0-367
Martine De Maziere :	martine.demaziere@aeronomie.be
Director General	Tel: +32-2-373-0-363

ASIMUT is the results of the work of many contributors:

- Nicolas Kumps
- Martine De Maziere
- Evelyn Dewachter
- Rachel Drummond
- Caroline Fayt
- Svetlana Kochanova
- Tobias Kerzenmacher
- Michel Kruglanski
- Vincent Letocart
- Alexis Merlaud
- Séverine Robert
- Ann C. Vandaele
- Sophie Vandenbussche
- Yannick Willame

ASIMUT RELEASE 11.0 : LICENSE AGREEMENT

Permission to use, copy, modify, and distribute any ASIMUT RELEASE 11.0, any documentation appertaining to this version of the VLIDORT, and any results obtained using this code is hereby granted without fee and without written agreement, provided that both the notice of copyright as expressed in this paragraph and the following two disclaimer paragraphs appear in all copies of the software.

IN NO EVENT SHALL IASB-BIRA BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THE ASIMUT RADIATIVE TRANSFER MODEL SOFTWARE AND ITS DOCUMENTATION, EVEN IF IASB-BIRA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOFTWARE IS WITH THE USER.

BECAUSE THE ASIMUT SOFTWARE IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING, THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IASB-BIRA HAS NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES ENHANCEMENTS OR MODIFICATIONS TO THE ASIMUT SOFTWARE.

As the ASIMUT Software is provided coupled to the (V)LIDORT SOFTWARE, please read also the (V)LIDORT licence agreement.